CEWES MSRC/PET TR/98-28

Grid Generation Capabilities Enhancement at the CEWES MSRC

by

Brian Jean
Richard Weed
Joe F. Thompson

DoD HPC Modernization Program
Programming Environment and Training
CEWES MSRC

CEWES MSRC

Nichols Research

04h02198

Work funded by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC 94-96-C0002
Nichols Research Corporation

GRID GENERATION CAPABILITIES ENHANCEMENT AT THE CEWES MSRC

Brian Jean * Richard Weed † Joe F. Thompson ‡

April 29, 1998

1 INTRODUCTION

Computational simulations based on finite difference, finite volume, or finite element schemes require either structured, unstructured, or hybrid computational grid systems that can contain millions of nodes for real world problems. Generation and refinement of these grid systems constitutes a large percentage of the total time required to perform the simulations. Although considerable amounts of research has been performed in the last 25 years to develop efficient methods for generating high resolution grids about complex geometries, grid generation continues to be a major pacing item in Computational Technology Areas (CTA) such as Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM). Other CTAs such as Climate, Weather, and Ocean (CWO) analyses and Environmental Quality Modeling (EQM) are also impacted. In spite of the fact that most of the available grid generation systems incorporate state-of-the-art mathematical models and computer graphics, the existing codes still require excessive amounts of human interaction and labor to produce acceptable grid systems for large scale analyses of complex geometries. These problems are intensified by the fact that current state of the art comprehensive grid generation systems are very large and complicated software systems that require considerable training and experience for effective use. The large learning curve of many of the most powerful of the current grid generation systems and the time constraints on generating solutions has prompted many users to stick with less capable but more familiar tools.

As outlined in a recent survey paper by Thompson [1], there is a general consensus among researchers and users that despite the advances that have been made in the last decade more automation needs to be incorporated into the grid generation process. Other areas where existing grid generation systems are known to be deficient are in the areas of interfaces to CAD systems or imbedding CAD capabilities into the system, solution adaption, coupling among different grid systems, and suitability for use in scaleable parallel computing systems. However, market forces are not large enough to encourage commercial development of grid generation systems to the level of sophistication attained by commercial CAD systems. Therefore, tools and enhancements that would reduce the time required to generate a grid system are not incorporated into the main software system and are left for the user to develop or find effective work-arounds to make up for the deficiencies.

It is recognized that no one grid tool will solve all the grid generation problems faced by users in the various CTAs. It is also recognized that the strengths and weaknesses of the different existing commercial and public

*Computer Engineer, CEWES-IH, USAE Waterways Experiment Station High Performance Computing Center, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199. E-mail: jean@sage.wes.hpc.mil

†CEWES PET CSM Onsite Lead, 1165 Porters Chapel Road, Vicksburg, MS 39180. E-mail: rweed@bohr.wes.hpc.mil

‡CEWES PET Academic Lead and Distinguished Professor of Aerospace Engineering, P.O. Box 9627, Miss. State, MS 29762. E-mail: joe@erc.msstate.edu

domain grid generation systems and their suitability for application in the users specific CTA are not known to the user community. Therefore, tools that might reduce a particular users grid generation time are not being used because the user is unaware of their existence. Also, many users have specific enhancements they would like to see in existing tools. Because of the importance of the grid generation process in computational simulations and the need to address the specific grid generation needs of the CEWES MSRC users, a Focused Effort was initiated under the CEWES MSRC Programming, Environment, and Training (PET) program to first evaluate existing commercial and public domain grid generation software and then use the information gained from the evaluations and from interactions with users to define a strategy for enhancing the grid generation capabilities for the both users of the CEWES MSRC and the Dept. of Defense in general. This white paper describes the work performed to obtain and evaluate a wide variety of commercial and public domain grid generation software that represent the currently available state of the art in grid generation.

2 OVERVIEW OF THE EVALUATION PROJECT

As discussed in the introduction, the overall goal of this project was to evaluate the capabilities of existing grid generation systems and use this information as a starting point to develop a strategy for defining and implementing user specific enhancements to the existing grid generation capabilities at the CEWES MSRC. The specific goals of the evaluation project were as follows:

1. Obtain and evaluate the most widely used commercial and public domain grid generation software used in the various CTAs.

2. Evaluate each software package to define qualities and characteristics such as its ease of use, relative strengths and weaknesses, the type of grid task the package is most suitable for, time required to generate an acceptable grid, etc.

3. Develop a set of benchmark geometries and grid tasks to be used in the evaluation process

4. Through user interaction, define a suite of desired enhancements and support tools that can be forwarded to the code developers or developed internally.

5. Educate users on the capabilities of the various grid generation codes.

6. Determine the applicability of the grid codes across the various CTAs

7. Sponsor a Grid Generation Capabilities Enhancement Workshop at which the results of the evaluation project would be presented and used as a basis for discussion of critical grid generation needs of CEWES MSRC users.

A suite of available grid codes was first defined for the various CTAs. These codes were broken down by topology (i.e. block-structured, unstructured hexahedral, unstructured tetrahedral, etc.). When available, evaluation copies of the commercial codes were obtained. The evaluation process then proceeded in two stages. The first stage was a learning phase in which the evaluator used the available documentation and sample cases to learn how to operate the code. The second stage was to apply the code to a set of benchmarks that were felt to represent the types of grid problems faced by the MSRC users.

A set of generic evaluation criteria were developed to measure code capabilities. These criteria were selected to give a subjective evaluation of the capabilities of individual codes from the viewpoint of a typical user. The impressions of the evaluator where tabulated in the forms given in Appendix A. A glossary of the terms used in the evaluation criteria is given in Appendix B. The grid evaluations, sample grids, and links to other sources of information on grid generation are available online at the following URL:

http://apollo.wes.hpc.mil/pet/CEWES/GRID/WORKSHOP/Grid_Enhancement.html

The codes evaluated to date are shown in the Evaluation Matrix given in Table 1 which precedes the Appendices. Only stand alone grid generation systems were selected for this evaluation. The results given in the Evaluation Matrix reflect our subjective rankings of different capabilities and features of the evaluated codes. No attempt has been made to evaluate grid generators embedded in flow or structural dynamics codes.

As anticipated, an initial survey of the available codes revealed that the majority of the existing grid generation tools have been developed to support CFD solvers. These codes are either block-structured codes or unstructured grid codes that produce tetrahedral or prismatic grids. The following list presents a brief description of each of the codes obtained during the course of the evaluation task.

- GRIDGEN: Gridgen is a GUI based block-structured grid code for CFD applications. Originally developed under an Air Force contract, the current version of Gridgen is a commercial product of Pointwise, Inc. Gridgen is used extensively in the aerospace community to generate grids about aircraft and missile geometries.

- GUM-B: Like Gridgen, GUM-B is an interactive GUI based block-structured grid code targeted at CFD applications. GUM-B is a research code developed at the National Science Foundation Engineering Research Center for Computational Field Simulations at Miss. State University. GUM-B is based on grid technologies developed for the National Grid Project.

- GRIDPRO: GridPro is a block-structured grid code developed Program Development Corporation (Peter Eiseman, et al.). The GridPro system consists of a suite of codes that combines a powerful scripting language with a GUI and an advanced grid smoothing algorithm. The system contains both batch and GUI components.

- CFD-GEOM: CFD-GEOM is an interactive grid code developed by CFD Research, Inc. It supports CAD type geometry creation and generates structured, unstructured, and hybrid grid systems. CFD-GEOM can generate grids for both CFD and Finite Element analyses.

- HEXAR: HEXAR is an automatic hexahedral grid generator developed by Cray Research. HEXAR requires a surface definition from CAD surface object description files. Once the surfaces are defined, the rest of the grid is generated with little or no interaction from the user.

- CUBIT: Cubit is an unstructured mesh generation tool for two and three-dimensional Finite Element analyses. Cubit generates quadrilateral and hexahedral grids using a combination of grid generation technologies. Cubit was developed at Sandia National Labs and is licensed without charge to government users.

- TRUEGRID: TrueGrid generates both unstructured and structured hexahedral meshes for Finite Element and CFD applications. TrueGrid is a commercial spinoff of the INGRID grid code distributed with the DYNA3D Finite Element code developed by Lawrence Livermore National Labs. TrueGrid extends the functionality of INGRID by offering a more interactive interface while still retaining the scripting capabilities and native support for DYNA3D provided by INGRID. TrueGrid is a product of XYZ Scientific Applications.

- SOLIDMESH: SolidMesh is an interactive unstructured grid code developed by the NSF ERC at Miss. State. SolidMesh generates triangular and tetrahedral meshes for CFD and Finite Element analyses. SolidMesh is a research code.

- VGRID/GRIDTOOL: VGRID generates unstructured tetrahedral meshes. The base code was developed by Vigyan, Inc. under NASA funding and is distributed free to U.S. users. VGRID uses a seperate GUI utility developed at NASA, GRIDTOOL, for surface definition, etc. VGRID is targeted at unstructured finite volume/finite element CFD applications.
- GEOMESH/X3D: GEOMESH is an unstructured (triangular or tetrahedral) Finite Element grid generation package developed at Los Alamos National Labs for geological applications. GEOMESH is built on the X3D (currently called LaGrit) collection of user callable grid generation and optimization tools developed at Los Alamos.

The INGRID code was already available at the CEWES MSRC. Site or single seat licenses were obtained for TrueGrid, VGRID/GRIDTOOL, HEXAR, SolidMesh, and GUM_B, Cubit, and LaGrit (Geomesh/X3D). Evaluation copies were obtained for the other codes.

3 Synopsis of the Grid Generation Enhacement Workshop

3.1 Background

A workshop on the utility of grid generation systems for MSRC users was held at the University of Texas in Austin on February 11-12, 1998. The workshop was attended by 42 researchers and managers from DoD, DoE, and participating universities. This workshop was targeted specifically at the five "grid-related" CTAs (CFD, CSM, CWO, EQM, CEA).

The purpose of this workshop was discovery and strategy to:

1. Identify the needs of CTA users that are not being met with currently available grid (mesh) generation systems.
2. Formulate strategy to work with grid code developers and/or vendors to meet those needs.

The development of a new grid generation package from scratch was specifically **not** a purpose of this workshop.

The mode of this workshop was evaluation and focused discussion with the following objectives:

1. Report at the Workshop the results of the evaluations of available grid generation systems of potential interest to CEWES MSRC users.
2. Report at the Workshop the capabilities of currently available geometry interfaces to grid generation systems.
3. Report at the Workshop the capabilities of currently available domain decomposition and other parallel considerations for grid generation.
4. Hear from the CTA users at the Workshop the grid-related needs that are not being met with grid generation systems now in use.
5. Through focused and directed discussion at the Workshop, formulate strategy to meet the grid-related needs identified.

The workshop was designed to broaden the awareness of the availability of grid generation resources in the MSRC user community.

The focus of the workshop was targeted at four specific gridding issues:

- CAD and other input interfaces.
- Adaptation driven by solution systems.
- Coupling among grid systems and with solution systems.
- Scalable parallel concerns, including decomposition.

3.2 Summary of Workshop Discussion and Conclusions

The following summary outlines the discussions held at the workshop and some of the conclusions from the various attendees. A seperate report will be issued that discusses the results of the workshop in more detail. The workshop was moderated by Professor Joe Thompson. Representatives of the DoD CTA leaders of the five "grid related" CTAs presented overviews of the status of grid generation as it impacts their particular CTA. After the presentation by the CTA representatives, the first day's discussion was focused on identifying a consensus on the deficiencies of current grid generation systems and how these deficiencies impact particular analyses. Some of the items discussed where:

- The need for software to generate very large grid systems for both CFD and CSM analyses on the current generation of parallel computers and support tools for analysis and visualization of very large data sets.
- The need for a standardized CAD interface to supplant the current reliance on IGES and more reliable CAD tools for "repairing" the CAD geometries prior to use in a grid generator.
- The need for more automation in existing systems.
- The need for more coordination among grid code developers and users to insure that needed capabilities can be implemented in a more timely manner.

The second day's discussion focused on defining possible solutions to current problems in grid generation. Among the approaches discussed were the development of "wrapper" codes that can be used to enhance the communication between different grid generation tools and to provide a more seamless environment for the grid generation process that spans from geometry definition to final grid.

Other discussion focused on the need for more cooperation between DoD, DoE, and university researchers. The need for alerting other users to the availability of "in-house" tools was emphasised.

The results from the discussions will be used by the PET team members to formulate a plan to develop a suite of tools to enhance the existing grid generation capabilities at the CEWES MSRC and to foster continuing communication with users and grid code developers.

4 RESULTS AND CONCLUSIONS

The results of the grid generator evaluation project and the Grid Generation Capabilities Enhancement Workshop can be summarized as follows:

- Current grid generation systems will generate acceptable grids about most geometries if the user is willing to spend the time required to refine the defining geometries and resulting grids.

- There is still not enough automation in the grid generation process.
- More emphasis needs to be placed on generation, analyses, and visualization of very large grid systems on parallel computing platforms.
- More interaction is needed between users and code developers in both DoD and DoE.
- More emphasis needs to be placed on the generation of large unstructured hexahedral grids of the type used in large scale CSM analyses.
- There is a definite need for more support tools for the existing grid codes and some mechanism for implementing a seamless grid generation process that encompasses all phases of grid generation and geometry definition.

References

[1] J. F. Thompson, "A Reflection on Grid Generation in the 90's: Trends, Needs, and Influences," *5th International Conference on Numerical Grid Generation in Computational Field Simulations*, Ed's, B. K. Soni, J. F. Thompson, J. Hauser, P. R. Eiseman, 1997.

Evaluation Matrix

Capability/Code	Gridgen	GridPro	GUM-B	GEOM	TrueGrid	INGRID	HEXAR	Cubit	SolidMesh	VGRID
Tetrahedral	NA	NA	NA	3	NA	NA	NA	NA	5	4
Hexahedral	4	5	4	2	3	3	2	3	NA	NA
GUI	4	2	5	3	3	NA	NA	NA	5	1
Script	2	5	NA	NA	4	4	NA	4	NA	NA
CAD Interface	3	2	4	2	3	1	1	Spec	4	3
Geom. Repair	3	2	4	3	3	NA	1	Spec	4	3
Automation	2	4	3	1	3	2	5	3	3	3
Stability	4	4	3	5	4	4	3	3	3	3
Analysis Support	4	3	1	2	4	2	1	Spec	2	??
Usability	4	4	4	3	3	2	4	3	4	3
Documentation	4	2	1	4	3	3	1	2	3	1
User Support	5	3	2	2	3	2	1	2	3	3
Cost	M	H	NA	M	M	NA	NA	NA	NA	NA
Overal	4	3	3	2	3	2	2	3	4	3

Definitions of Terms
[1=Lowest, 5=Highest, NA=Not Applicable, Spec=Special Case Not Easily Rated, ??=Unknown]

Table 1. Evaluation Matrix

APPENDIX A
GRID GENERATOR EVALUATIONS

A.1 Evaluation Summary for Gridgen

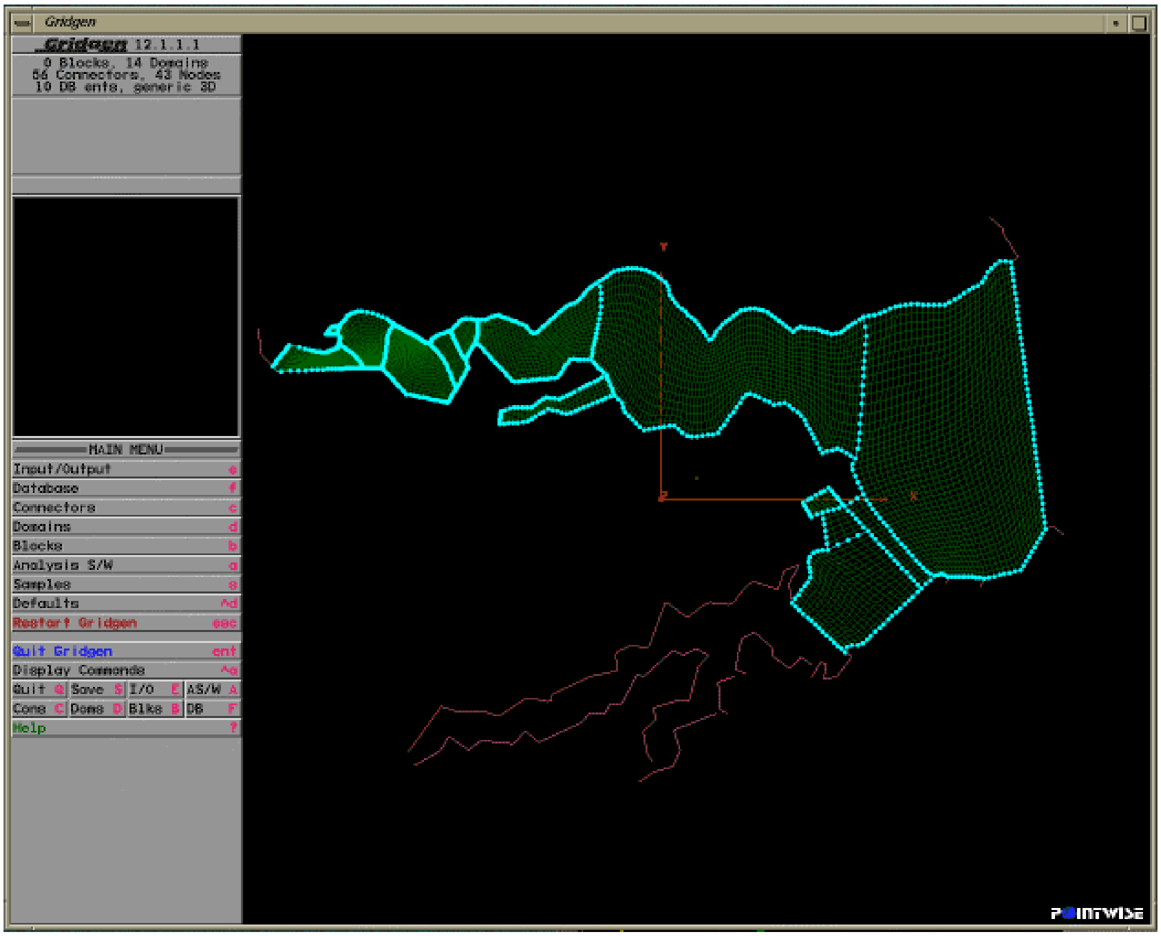

Figure 1: GRIDGEN Interface and Sample Grid

```
Software Title:  Gridgen
Version: 12
Vendor: Pointwise
Supported Platforms:  SGI, IBM, HP
Availability: Commercial Product
Contact:
          Pointwise, Inc.
          PO Box 210698
```

Bedford, Texas 76095-7698
USA

Phone: 817.377.2807
Toll-free: 888.GRIDGEN
Fax: 817.377.2799
Email: gridgen@pointwise.com
URL: http://www.pointwise.com

Documentation Provided: User Manual with tutorials
User Support Hotline: Yes

Reviewer(s): Brian Jean, et al.

Grid Types: Multi-Block Structured, Unstructured Hexahedral
Grid Input Formats: Plot3D
Grid Output Formats: Plot3D, Proprietary Save State, +supported analysis
codes listed below.
Geometry Input Formats: IGES, PATRAN (packets 32 and 33 only), and
Proprietary Database
Geometry Output Formats: Proprietary Database

Supported Analysis Codes: CFX-4, GASP, VSAERO, INCA, TASCflow, PHOENICS,
FLUENT, FLUENT/UNS, RAMPANT, Star-CD, TEAM,
NPARC, OVERFLOW, CNSFV, DTNS, FANS, FALCON, ADPAC,
COMO
*Generic format also provided and documented

See Appendix B for an explaination of the items below

Automation Level:	Low
Seat Time for Infrequent User:	High
Seat Time for Frequent User:	High
User Interaction Paradigm:	Good
User Interface Type:	GUI, modal, some commands can be scripted
Learning Curve:	Easy
Documentation Quality:	Good
User Support Quality:	Good
Overall Capability:	Good
Grid Quality:	Good
Curve/Surface Grid Fidelity:	Good
Large Data Set Capability:	Good
Stability/Maturity:	High
Support for MPP Environment:	Planned for future versions
Grid Editing Features:	Reactive gridding via replacement

```
                                  of database entities.
Quality Measure Evaluation:       Negative Jacobian check only.

------------------------------------------------------------------------------------------
```

Code Summary:

Gridgen is a solid, very functional hexahedral grid generation program tailored to block-structured grids. The mouse-driven graphical interface is modal, which allows the code to lead the user through most operations. This is an advantage to the novice or infrequent user, but may hinder the power user as it typically requires a relatively large number of user actions to complete an operation. The interface is relatively clean, the layout is logical, and it is easy to navigate between modules and control panels. Documentation and user support is very good.

Geometry is typically imported via an IGES or PATRAN Neutral file. Gridgen also provides a proprietary composite database format as a save state for geometry information, since neither IGES nor PATRAN files can be exported. Geometry can also be constructed from scratch using the internal CAD system.

Grid topology is specified using "connectors" to form "domains" or faces which are, in turn, used to form blocks. Connectors may reside in space or be constrained to lie on a curve contained in the database. Domains (or surface grid patches) are formed by specifying a network of connectors which form the four logical edges of the domain. Note that multiple connectors may form a single logical edge of a domain. Blocks are formed by quilting domains to form the six logical faces of the block. Multiple domains may form a single logical logical face of a block.

The number of points and the point spacing within the grid are controlled using the connectors. A variety of spacing functions are available to control initial point distributions on the connectors. Additional control of point distribution is available through the domain elliptic smoother, which provides an extensive set of control features.

Grid points on connectors are constrained to lie on the database curves to which the connectors are constrained, if any. Domain grids may be generated in space, projected to a surface, or generated within the parametric space of a surface. Domain grids can be generated with a wide range of algebraic solvers. An excellent surface elliptic solver is available and provides a large number of control functions for smoothness and orthogonality. Volume grids are generated via transfinite interpolation, but no volume elliptic smoothing is available within the code, however, the code will write an input deck for GRAPE3D volume elliptic smoother from NASA Ames Research Center and Gridgen3D.

The current version of Gridgen has only basic grid visualization capability. Surface grids may be viewed as wireframe, wireframe with hidden lines removed, and shaded surfaces. Volume grids are visualized with plane stepping. No quality measures are available.

A boundary condition editor allows analysis code BCs to be set within Gridgen. Boundary conditions may be set for any of the supported codes listed above and exported into a boundary condition file. Grids may be exported in a variety of Plot3D formats as well as formats compatible with any of the supported codes listed above.

A subset of Gridgen's interactive commands can be scripted in a GCL file and used to run Gridgen in batch mode.

Reviewers Comments:

- Original domain interfaces cannot be recovered if they are allowed to float during surface grid smoothing.
- Most operations are tedious, requiring several mouse clicks to complete. This is partially due to the modal interface employed by the code.
- The grid generation process is well defined and easy to follow. The code often prompts the user to do the next required operation.
- Construction of faces (domains) and blocks is completely manual as is setting the orientation (index direction) of blocks.
- When setting the orientation of a block, the code will let the user build a left-handed coordinate system (the code does produce a warning).
- The elliptic solver for faces (domains) is excellent. It provides very fine control over grid behavior, boundary conditions, off-boundary spacing and boundary orthogonality.
- Many algebraic grid options are provided.
- No elliptic volume smoothing.
- Display options are awkward and sometimes confusing.
- Curve on surface feature is hard to use.

Below are comments received other users:

Pros:

- Overall, a very good general purpose grid generator.
- Excellent customer service (will make rapid modifications to the code to fix our specific problems)
- Good import/export capability (does well with IGES, etc.)
- Complete projection capability (linear, closest point, cylindrical, spherical, etc.)
- Excellent elliptic solver (good control over solver parameters)
- Good GUI - very easy to use once user is familiar with UI.
- Has keyboard equivalents to mouse actions.
- Very good copy, translate, scale, and rotate capability.
- Good automatic generation of edge connectors and good tools for generation of surface domains and blocks.
- Good grid visualization tools.
- Complete cell volume checker and a good negative or skewed volume view capability.
- Reasonably stable.

Cons:

- Gridgen terminates with no warning if the license server goes down - very annoying.
- File I/O is slow. Always converts to and from GG format when importing or exporting, bogging down the system.
- Cannot change spacing on a series of connectors. User must currently select and change connectors one at a time.
- Cannot automatically create faces and blocks (not a big problem but would be nice to have)
- Cannot convert Gridgen connectors and domains into equivalent database entities. This a very significant shortcoming.

All curve edits have to be done in the plane of the screen. In other words, in the world coordinate system, if you want to change the curvature of a line in the world XY plane, the geometry must be rotated into the screen XY. This is no problem if you need to edit within the XY, XZ, or YZ planes but its difficult to edit curves if the edit plane crosses all three dimensional axes. There is no way to type in rotation angles for precise orientation of the geometry for editing curves.

A.2 Evaluation Summary for GUM-B

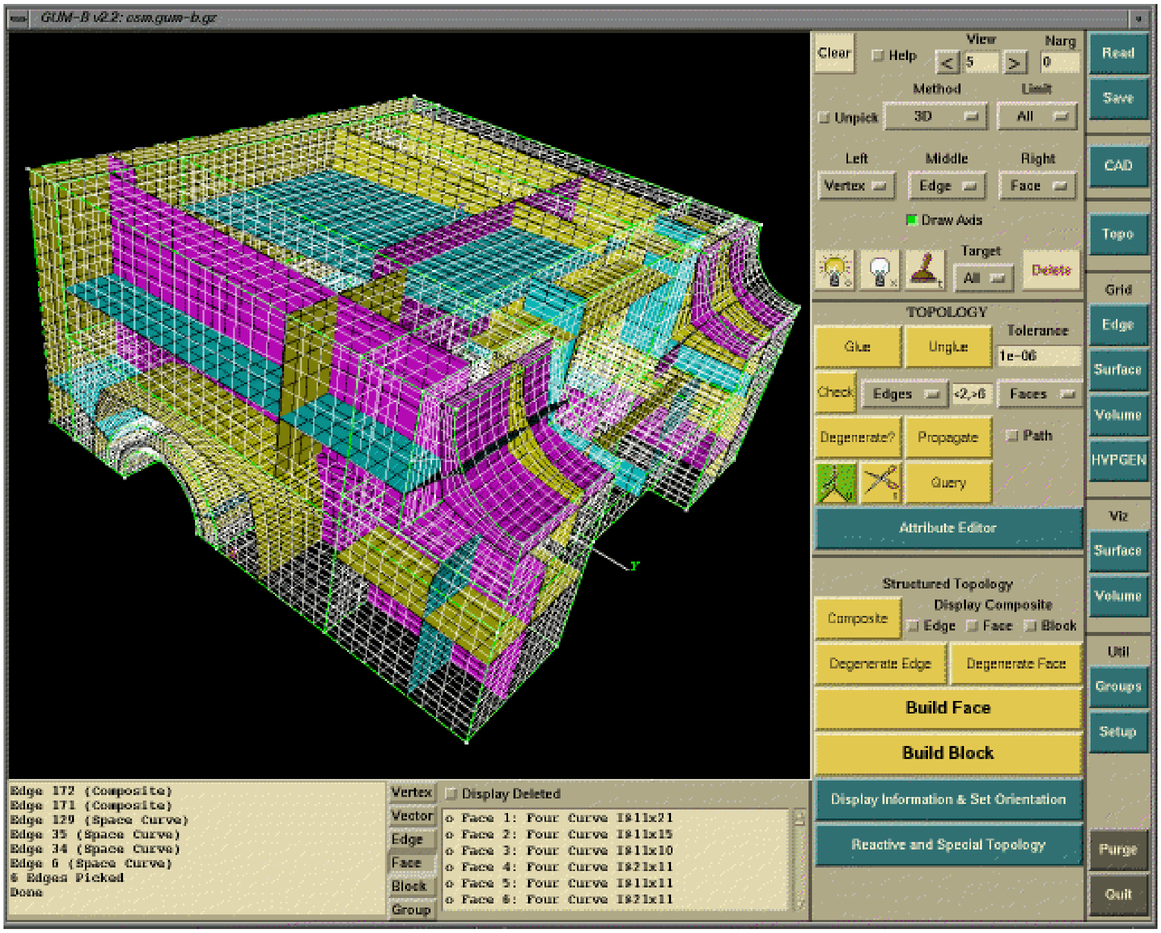

Figure 2: GUM-B Interface and Sample Grid

```
Software Title:  GUM-B
Version: 2.2
Vendor: Mississippi State University
Supported Platforms:  SGI
Availability: Special (Contact MSU/NSF ERC for CFS)
Contact:
           Mike Remotigue
           Engineering Research Center
           PO Box 9627
           Miss. State, MS 39762
           USA

           Phone: 601.325.8278
           Fax: 601.325.7692
```

Email: remo@erc.msstate.edu
URL: http://www.erc.msstate.edu

```
Documentation Provided:  Online tutorials
User Support Hotline:  Yes (software authors)

Reviewer(s): Brian Jean

Grid Types: Multi-Block Structured, Unstructured Hexahedral
Grid Input Formats: Plot3D, EAGLE
Grid Output Formats: Plot3D, Proprietary Save State, Unstructured Hex.
Geometry Input Formats:  IGES, StereoLithgraphy, Structured and/or unstructured
                         surface grids, generic discrete curve and surface
                         data.
Geometry Output Formats: IGES, stereo lithography

Supported Analysis Codes: Generic format for BC setup

---------------------------------------------------------------------------------------
```

See Appendix B for an explaination of the items below

```
Automation Level:                 Medium
Seat Time for Infrequent User:    High
Seat Time for Frequent User:      Medium
User Interaction Paradigm:        Good
User Interface Type:              GUI, modaless, no scripting/journaling
Learning Curve:                   Medium
Documentation Quality:            Poor to non-existent
User Support Quality:             Good

Overall Capability:               Good
Grid Quality:                     Good
Curve/Surface Grid Fidelity:      Good
Large Data Set Capability:        Good
Stability/Maturity:               Research/Academic Code


Support for MPP Environment:      None
Grid Editing Features:            Reactive topology snaps grid to new or
                                  relocated geometric entities.
Quality Measure Evaluation:       Skew, Aspect Ratio, Stretching,
                                  Off-Boundary spacing violations, Negative
                                  Jacobian Checking with Mulit-Grid
                                  Capability.

---------------------------------------------------------------------------------------
```

Code Summary:

Within GUM-B, grids are defined geometrically by curves and/or surfaces which can be imported and/or generated from scratch using the internal CAD system. Grid topology is defined by Edges, Faces and Blocks. Grid topology is completely independent of geometrical curve and surface topology, with geometric fidelity being maintained by projecting Face grids to their respective surfaces. The user employs the internal CAD system to build a wireframe representation of the blocking topology (i.e. a network of curves which define the Block Edges). After the wireframe has been constructed, the code will automatically determine all valid Faces and Blocks. The user may, but is not required to, set orientations of the Faces and Blocks. During this phase, the user is not required to specify grid information, such as numbers of points, and is not required to keep track of geometrical orientations. After the blocking topology has been specified and all Faces and Blocks have been built, the user begins specifying grid information. GUM-B automatically propogates dimensions throughout the grid, hence the user need not worry about insuring matching dimensions. Distributions can also be copied and/or propogated. Surface and volume grids are generated via transfinite interpolation and may be elliptically smoothed with either fixed or floating interface boundaries and a variety of control functions. The grid are visualized using several techniques including plane stepping and weather map display of various grid quality metrics. The code provides for specification of analysis code boundary conditions via a generalized format. Grids are exported via any number of Plot3D format variations or a generalized unstructured hexahedral format.

The code has an excellent user interface which consists of a graphics window, message window, entity list window, and an application dependent function panel. Function panels exist for CAD, topology, grid generation, visualization, grouping, and setup. The various panels are selected via buttons along a side bar. A unique feature of this code is the availability of both left-handed and right-handed interface layouts (a real advantage for you creative types). There are many time-saving features available. The most obvious and probably most significant of these is the modaless interface. To perform an operation, the user selects the input for the desired operation and then clicks on the button to envoke it. This eliminates the need to prompt the user for input and drastically reduces the amount of time required to complete a particular operation. In the hands of an experienced user, operations can be performed very efficiently and with relatively few mouse-clicks and keystrokes. Another important user interface feature is overloaded function buttons. Most operations within the code behave differently depending on the type of input given. In most cases this allows the user to do the desired operation in the most efficient manner under the given circumstances. For example, a line can be created between two existing points, or from one point, a direction vector, and a distance, or a series of line segments can be created joining a two or more selected points; all using differing inputs to a single function button.

Reviewers Comments:

- User interface is clean and easily navigated.
- Many time saving features and short cuts for the experienced user.
- User interaction paradigm is consistent throughout.
- Good grid visualization/quality measure features.
- Internal CAD system allows imported geometry to be repaired and/or modified. A complete geometry database can be created from scratch.
- The availability of the code and a pricing structure have not been set.

- No formal user support structure exists, although current users may contact the developers directly.
- Documentation is non-existent and the online tutorials are generally not revised to keep up with code changes.

A.3 Evaluation Summary for CFD-GEOM

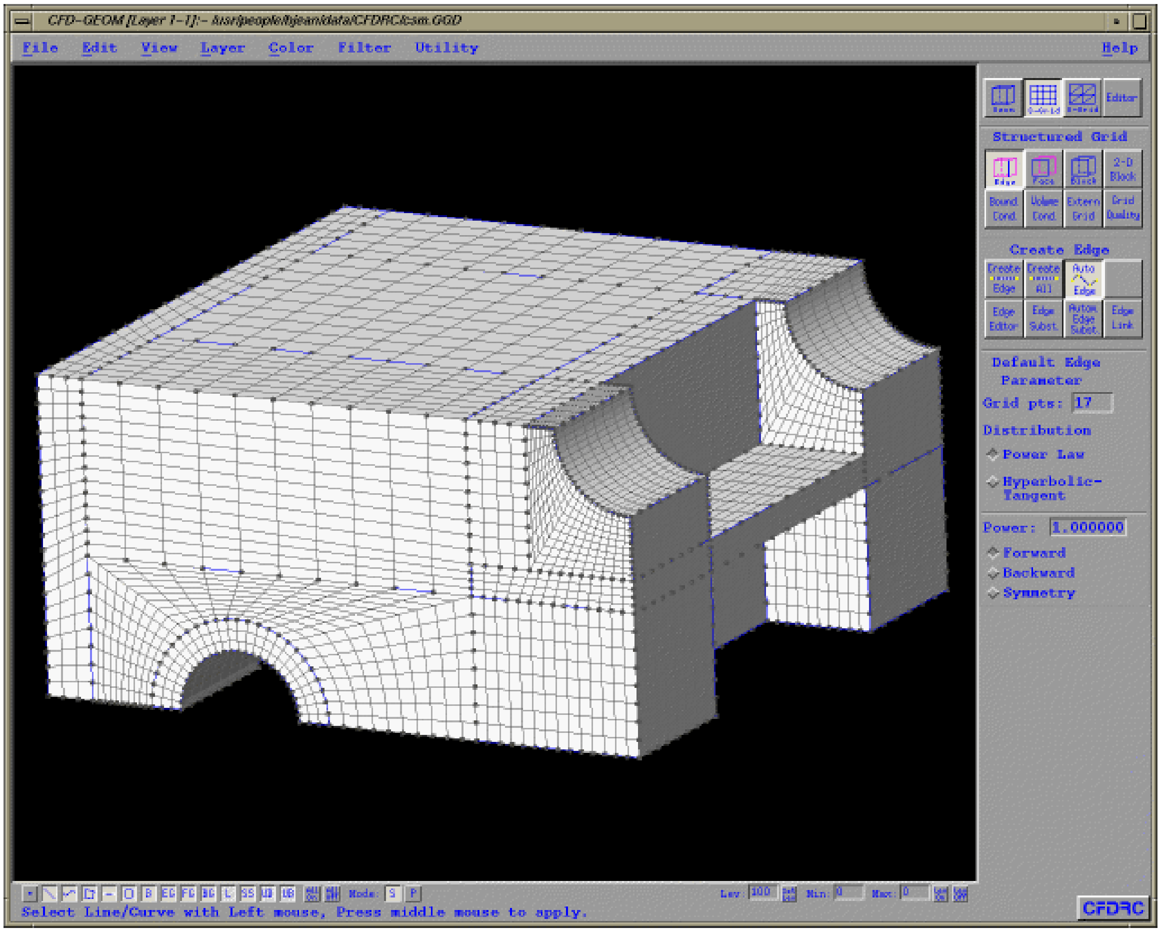

Figure 3: CFD-GEOM Interface and Sample Grid

```
Software Title:  CFD-GEOM
Version: 3.0
Vendor: CFD Research Corporation
Supported Platforms:  SGI, HP, IBM RS/6000, DEC, Sun, Windows95/NT
Availability: Contact CFDRC
Contact:
           R.Sukumar
           CFD Research Corporation
           Cummings Research Park
           215 Wynn Drive
           Huntsville, AL 35805
           USA

           Phone: 205.726.4800
```

```
          Fax: 205.726.4806
          Email: sales@cfdrc.com
          URL: http://www.cfdrc.com

Documentation Provided:  Users manual with tutorials
User Support Hotline:  Yes

Reviewer(s): Brian Jean

Grid Types: Multi-Block Structured, Unstructured Hexahedral,
            Unstructured Tetrahedral, Mixed Element.
Grid Input Formats: Plot3D, NASTRAN, PATRAN, StereoLithography, FAST,
                    CFDRC Mixed Element Format (MFG).
Grid Output Formats: Plot3D, FAST, NASTRAN, PATRAN, CFDRC Mixed Element
                     Format (MFG), CFDRC Data Transfer Facility
                     format (DTF).
Geometry Input Formats:  IGES, Plot3D, Proprietary
Geometry Output Formats: IGES, Proprietary

Supported Analysis Codes: CFD-ACE, User configurable boundary condition
                          and initial condition files.

-----------------------------------------------------------------------------------------
```

See Appendix B for an explaination of the items below

```
Automation Level:                 Low (totally manual)
Seat Time for Infrequent User:    High
Seat Time for Frequent User:      High
User Interaction Paradigm:        Average
User Interface Type:              GUI Mixed-Mode Modal/Modaless depending
                                  on the operation being performed.
Learning Curve:                   Moderate to Easy
Documentation Quality:            Average (structured grids), average
                                  for unstructured package.
User Support Quality:             Average

Overall Capability:               Average to Poor
Grid Quality:                     Average to Poor
Curve/Surface Grid Fidelity:      Good
Large Data Set Capability:        Average (they define a typical ''large''
                                  data set as 5 to 10 million nodes)
Stability/Maturity:               Commercial Product


Support for MPP Environment:      None
Grid Editing Features:            Geometry editing and/or replacement
                                  updates attached grid and geometry
                                  entities.
Quality Measure Evaluation:       Jacobian check, skew angle, aspect
```

```
ratio for structured grids and face
angle and point-normal check for
unstructured grids.
```

Code Summary:

CFD-GEOM is tailored for use within CFD-ACE, the software environment furnished by CFDRC, however it can also be used as a stand-alone grid generator. It is a very labor-intensive code that requires a lot of user interaction and provides very little automation of tasks. However, it is quite stable and the functions available work well.

CFD-GEOM has a good user-interface that is reasonably easy to learn and manipulate. The only noticable problem was difficulty in manipulating objects for view, rotation in particular was difficult to use effectively. The function buttons and menus are well placed and easy to understand and learn.

The code was accompanied by a complete set of documentation, which included a set of tutorials and users manuals for all codes in the CFDRC software suite (each code has its own seperate manual). The original package did not include any unstructured grid tutorials, however when I contacted software support, I was told that they were available and they shipped them to me via overnight mail. Overall quality of the documentation is good. It is well organized and easy to read and understand.

The internal CAD system is quite good and very functional. Simple to moderately complex geometries can be constructed directly with CFD-GEOM and there are several tools which facilitate repair of imported geometries. Trimmed surfaces are supported. Geometries not created in CFD-GEOM may be imported via IGES or as a structured grid

Structured grids are built by first constructing face edges on the existing geometry. Edges and edge sets are used to construct faces, and faces and face sets are used to construct blocks. Partial face matching is supported for both faces and blocks, and blocks may be joined to create composite blocks. The user must keep track of the number of points on each edge to insure that a face or block is valid. This is a serious shortcoming when dealing with the very complex blocking topologies required for most "real world" problems. Control of grid orientation is provided and once a block orientation is set, it can be propogated to adjoining blocks. The strongest feature of CFD-GEOM is grid editing. CFD-GEOM tracks relationships between grid entities and geometry entities. This allows new geometries to be substituted for old ones while keeping the same grid topology. Existing geometry can also be edited. In either case, the grid will snap to the new or edited geometry and changes will propogate as necessary throughout the grid structure. The most serious shortcoming of the code is its lack of grid quality control features. Face and/or block interfaces must be smoothed by editing their shape (they will not "float" as with most elliptic smoothers) and the grid smoothing that is available is rudimentary at best.

Reviewers Comments:

- Only a partial evaluation was done using the structured grid capability. The unstructured grid capabilities were not evaluated.
- Rotation of objects for viewing is difficult to control.
- Tear off menus are a nice feature
- The code does not automatically track orientation of grid objects and will allow the user to create a left-handed system. The code does warn the user when a left-handed system is generated.

- Volume grids can be oriented as the user desires, however there is no way to determine the current orientation of a block and the reorientation tool can be confusing to use. The code can propogate orientation changes to neighboring blocks (this is a very nice feature).
- Time-saving features in the user interface are nice, particularly in the CAD system.
- Good documentation with a good suite of tutorials. However, some of the tutorials can be hard to follow and no tutorials are given for the unstructured grid generator.
- The Virtual Parts Library is a nice feature.
- The code seems to stress geometry creation and geometry/grid editing more than grid quality. For example, there seems to be very little control over grid smoothing for structured grids and relatively few point distribution features.
- Linking of grid entities for the purpose of propogation of grid information (number-of-point and spacing) changes is manual and explicit. After the user has specified links, the code will automatically propogate grid changes to linked entities.
- This code has the least automation of any structured grid code tested.
- Grid quality control for structured grids is poor to non-existent. No good smoothing functions are available. Improving grid quality is accomplished through editing of point spacings and topology and is very tedious.
- There is no way to directly create a curve which lies on a surface (i.e. in the parametric space of the surface). Curves on surfaces must be created in 3-D and projected or they must result from surface-surface intersections.

A.4 Evaluation Summary for HEXAR

```
Software Title:  Hexar
Version: 2.4
Vendor: SGI/Cray (see Reviewers Comments below)
Supported Platforms:  SGI (all platforms w/IRIX 6.2>),
                      all Cray PVP systems (YMP, J90, T90,
                      UNICOS 8.0 or 9.0).
Availability: Special (Contact SGI/Cray)
Contact: SGI/Cray
Documentation Provided:  Handouts from short course (available from
                         your onsite Cray Analyst).
User Support Hotline:  None.  Cray requires a person on-site to attend
                       a two day training course in order to support
                       users at the site.

Reviewer(s): Brian Jean, David Sanders

Grid Types: Unstructured Hexahedral
Grid Input Formats: None
Grid Output Formats: HyperMesh
Geometry Input Formats:  StereoLithography, Proprietary Triangulated
                         Surface Definintion
Geometry Output Formats: None

Supported Analysis Codes: None in native format.  Only generic unstructured
                          hex format is provided.
Quality of Documentation:  Below Average (a set of handouts and flow charts
                           from a training course).
Quality of User Support:   Below Average to poor (no access to software
                           authors).

-----------------------------------------------------------------------------------------
```

See Appendix B for an explaination of the items below

```
Automation Level:                 High
Seat Time for Infrequent User:    Low
Seat Time for Frequent User:      Low
User Interaction Paradigm:        Good
User Interface Type:              Command line
Learning Curve:                   Average

Overall Capability:               Good (For certain environments, see below)
Grid Quality:                     Good to poor
Curve/Surface Grid Fidelity:      Poor
Large Data Set Capability:        Good
Stability/Maturity:               Research Code
```

```
Support for MPP Environment:     None (Runs on Cray PVP machines)
Grid Editing Features:           None
Quality Measure Evaluation:      None
```

Code Summary:

Hexar is a highly automated set of codes for generating unstructured hexahedral grids. It is particularly useful when geometries are very difficult to visualize or have many hidden cavities. Examples include automotive underhood, biomedical, casting, and H-Vac systems. The code is less effective where a high quality mesh is required, or when a geometry is relatively easy to mesh by hand.

Input for Hexar is a Triangulated Surface Definition (TSD) file and various flags given on the command line. Several filters are provided to translate various geometry definitions into TSD format (the most reliable translator is for StereoLithography). For the code to operate properly, the surface definition must be closed (with the exception of small gaps which can be tolerated), and the surface must represent the outer boundary of the volume mesh. The surface definition should not have any internal baffles, or multiple material types. A large number of command-line options are available. A partial list appears below, however many other options exist.

- -restart_write: Writes a restart file at various checkpoints during exectution.
- -restart_read: Read the latest restart file and starts exection from the last checkpoint.
- -tsd: Input file name.
- -s: Smallest cell side allowed in the mesh.
- -ztl: Largest known gap or overlap in the TSD.
- -S: Largest cell side allowed in the mesh.
- -nc: Minimum number of cells between two opposite walls.
- -angle: Threshold value that determines a feature edge (0¡=angle¡=180).
- -da: Curvature influence.
- -lap: Number of Laplace smoothing passes.

File translators include:

- ENSIGHT to TSD
- IDEAS to TSD
- NASTRAN to TSD
- PATRAN to TSD
- SLA to TSD

- TSD to IDEAS
- TSD to PATRAN
- TSD to SLA

After completion, Hexar dumps a "map" file which must be processed by another code called "hexarpost." This code smooths the boundary surfaces, makes minor grid topology changes, and smooths the volume grid all to improve overall grid quality. Hexar-post is executed by a single command, again with various flags to control processing. Both codes can be executed in batch mode and queued in the NQS. No graphical user interface is provided and third-party software is required for visualization of the output grid. Currently the best choice for a third-party package is HyperMesh.

Reviewers Comments:

- This code is still under development and is not considered a "product" by Cray. It is still very much a research code.
- Hexar meshes tend to be more dense than manually generated grids.
- Very little control over mesh spacing and density.
- Hexar has great potential to drastically decrease the grid generation time for analysis where the objectives are quick turn-around time and "ball park" or quick and dirty answers. Poor geometric fidelity and relatively low grid quality may present problems in situations where a high accuracy solution is desired.
- The code is highly automatic and will produce a mesh on practically anything. However, some tuning may be necessary to get a mesh with acceptable quality.
- Third party software (namely Hypermesh) is required in order to prepare CAD geometry for processing and to visualize the grid and assess grid quality. A third party structured grid generator is also required in order to "tune" some features of the grid.
- The most reliable surface definition for translation to TSD format is StereoLithography.
- No support is provided for boundary condition setup or material property assignment.
- Hexar is both memory and cpu intensive, however it is almost completely automatic.
- A commercial version with integrated GUI and visualization is currently being developed by independent software company owned by the original author of Hexar.

A.5 Evaluation Summary for Cubit

```
Software Title: Cubit
Version: 2.0.3
Vendor: Sandia National Laboratories
Supported Platforms: HP, SGI, Sun
Availability: Contact Sandia National Laboratories
Contact:
           Marilyn K. Smith
           Technology Programs Department
           Division 1503, MS-0833
           Sandia National Laboratories
           P.O. Box 5800
           Albuquerque, NM 87185-0833
           USA

           Phone:
           Toll-free:
           Fax: 505-844-9297
           Email: mksmith@sandia.gov
           URL: http://endo.sandia.gov/SEACAS/CUBIT/Cubit.html

Documentation Provided: User manual with tutorials
User Support Hotline: None

Reviewer(s): Brian Jean, Rick Weed

Grid Types: Unstructured Hexahedral
Grid Input Formats:ExodusII
Grid Output Formats: ExodusII
Geometry Input Formats: ACIS SAT, PRO/Engineer via PRO/E to ACIS
                        Translator, FASTQ
Geometry Output Formats: ACIS

Supported Analysis Codes: All SECAS Codes


-----------------------------------------------------------------------------------------
```

See Appendix B for an explaination of the items below

```
Automation Level:                 Medium
Seat Time for Infrequent User:    High
Seat Time for Frequent User:      High
User Interaction Paradigm:        Good
Interface Type:                   Command line or script with graphics
                                  window and limited mouse manipulations.
                                  Batch mode available.
```

```
Learning Curve:                  Moderate
Documentation Quality:           Good
User Support Quality:            N/A

Overall Capability:              Average
Grid Quality:                    Average
Curve/Surface Grid Fidelity:     Good
Large Data Set Capability:       Good
Stability/Maturity:              Average/Research Code

Support for MPP Environment:     None
Grid Editing Features:           Grid editing via geometry replacement
                                 and/or script editing.

Quality Measure Evaluation:      Aspect ratio, skew, taper, stretch,
                                 diagonal ratio, cell area/volume, and
                                 warpage displayed as colored weather
                                 map, histogram, or numerical values
                                 in table form.

-----------------------------------------------------------------------------------------
```

Code Summary:

Cubit is a semi-automated mesher which is tailored for use with solid models in ACIS and PRO/Engineer format. The code can directly import ACIS models and PRO/Engineer models which have been converted to ACIS SAT format. Simple geometries can be constructed and meshed directly within Cubit. The geometry engine is based on the ACIS solid modeler and provides basic solid primatives which include Brick, Pyramid, Cylinder, Prism, Frustum, Torus, and Sphere. These primatives may be copied, scaled, translated, rotated, and reflected. In addition, basic CSG operations such as intersection, subtraction, and union can be performed. A sweep operation is also provided to allow construction of more complex solid bodies. Cubit geometry can be exported in ACIS SAT format.

Mesh entities are associated with a geometry entity which owns it. This allows geometry properties to be automatically propogated to the mesh. For example, setting an attribute on a surface affects all mesh entities owned by that surface. Point distribution within the mesh is controled by specifying mesh density and by biasing point spacings on curves. A boundary layer tool allows very tight control over boundary layer regions. The user can specify the spacing of the first grid line off the surface, a growth factor, and either the number of cells or a total depth. When meshing begins, the code attempts to automatically match intervals between adjacent surfaces within constraints defined by the user. This feature can be overridden by the user if needed. Entities may be meshed using a variety of schemes. Currently supported schemes are as follows:

For Curves:

- Equal: Linear distribution of points
- Biased: Clustering of points to one curve end with stretching defined by a growth factor
- Featuresize: Clustering based on geometric features of the curve.

For Surfaces:

- Map: Structured surface mesh
- SubMap: Auto-decomposition of a surface into subregions to produce an overall structured mesh.
- TriMap: Generates triangular elements at sharp corners or specified vertices and generates a structured mesh on the remainder of the surface.
- Pave: Advancing front method for general surfaces including trimmed surfaces (Paving).
- TriPave: Generates triangular elements at sharp corners or specified vertices and paves the remainder of the surface.
- Triangle: For meshing three-sided regions.

For Volumes:

- Map: Structured volume mesh.
- SubMap: Auto-decomposition of the volume into subregions to produce an overall structured volume mesh.
- Project: 1/2D sweeping along a general path (accepts draft angles).
- Translate: 1/2D sweeping along a vector.
- Rotate: 1/2D sweeping about a central axis (with non-zero inner radius).
- Plaster: Research algorithm (Plastering)
- Weave: Research algorithm (Whisker Weaving).

For the purpose of specifying boundary conditions, points and cells can be grouped into Nodesets, Sidesets, and Blocks. Nodesets and Sidesets are defined by specifying the curves and/or surfaces which make up a Nodeset and/or Sideset. Nodes may belong to multiple Nodesets.

Surface and volume meshes can be smoothed using either equipotential, laplacian, or centroid area pull functions. A variety of stencils are available for the default equipotential option. Several surface and volume mesh quality metrics can be evaluated. Aspect ratio, skew, taper, and stretch are available for both surface and volume meshes. In additition, warpage, and element area are available for surfaces and cell volume and diagonal ratio are available for volumes. Quality metrics can be displayed in numerical form, as a histogram, or as a color "weather map" on the grid itself.

Meshes are exported via the ExodusII file format. Documentation on SEACAS codes as well as information on the ExodusII file format can be found on the Cubit web page at the URL given above.

Reviewers Comments:

- The plastering and weaving algorithms for volume meshing are still experimental and not particularly reliable. Use them with caution.
- The automated interval matching for surfaces is a nice feature, but requires a good deal of experience to avoid unexpected results.
- The scripting language is consistent and easy to learn.
- The ability to edit a script and replay it is very nice.
- The command-line interface is cumbersome, especially when trying to specify a particular entity as the target for a command. The graphics window helps this situation, but is not a substitute for a well designed GUI.
- The available gridding algorithms are robust and generally yield good quality grids (except as noted above).
- Although the code offers some automation, the user can still have good control of mesh spacing and topology.
- A nice feature of the surface paving algorithm is the ability to adaptively mesh a surface based on a variety of element sizing functions.
- Elements can also be sized based on a general field function read from an ExodusII file.

A.6 Evaluation Summary for SolidMesh

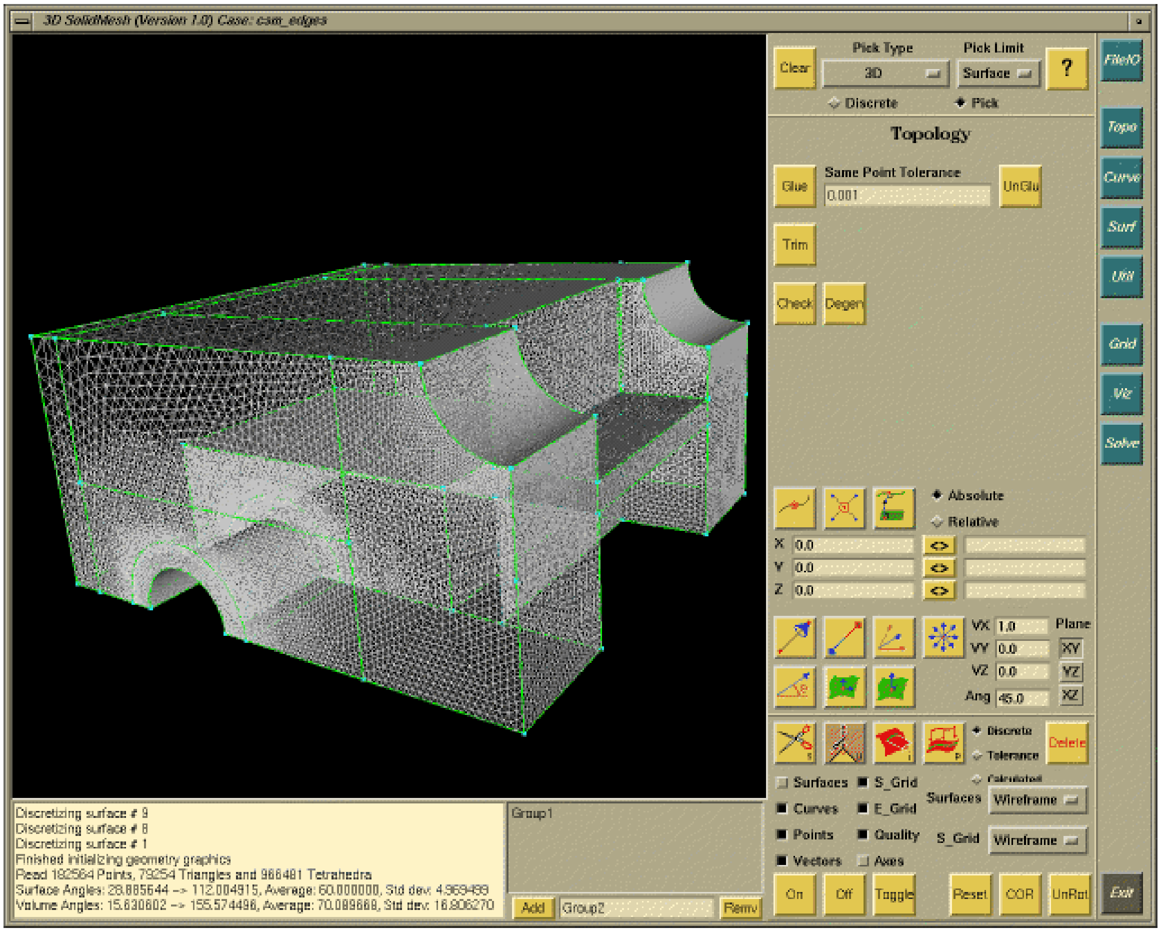

Figure 4: SolidMesh Interface and Sample Grid

```
Software Title:  SolidMesh
Version: 2.0
Vendor: Mississippi State University
Supported Platforms:  SGI
Availability: Special (Contact MSU/NSF ERC for CFS)
Contact:
          Dr. Dave Marcum
          Engineering Research Center
          PO Box 9627
          Miss. State, MS 39762
          USA

          Phone: 601-325-8278
          Fax: 601-325-7692
```

```
          Email: marcum@erc.msstate.edu
          URL: http://WWW.ERC.MsState.Edu/thrusts/grid/solid_mesh/index1.html

Documentation Provided:  Online users manual with tutorials
User Support Hotline:  Yes (software authors)

Reviewer(s): Brian Jean

Grid Types: Unstructured Triangle/Tetrahedra
Grid Input Formats: FAST
Grid Output Formats: FAST Unstructured, proprietary format.
Geometry Input Formats:  IGES, structured surface grids,
                         generic discrete curve and surface data.
Geometry Output Formats: IGES

Supported Analysis Codes: MSFENS

------------------------------------------------------------------------------------------
```

See Appendix B for an explaination of the items below

```
Automation Level:                 Medium to High
Seat Time for Infrequent User:    High
Seat Time for Frequent User:      Medium to Low
User Interaction Paradigm:        Good
User Interface Type:              GUI, modaless, no scripting/journaling
Learning Curve:                   Moderate
Documentation Quality:            Average
User Support Quality:             Good

Overall Capability:               Good
Grid Quality:                     Good
Curve/Surface Grid Fidelity:      Good
Large Data Set Capability:        Good
Stability/Maturity:               Research/Academic Code


Support for MPP Environment:      None
Grid Editing Features:            None
Quality Measure Evaluation:       Surface angle weathermap, volume
                                  angle weathermap, and surface
                                  solid angle.

------------------------------------------------------------------------------------------
```

Code Summary:

Within SolidMesh, grids are defined geometrically by curves and/or surfaces which can be imported and/or generated from scratch using the internal CAD system. The code is capable of importing IGES data including trimmed surfaces, Plot3D surface grids, generic ordered discrete curve and surface data, and unstructured surface grids.

Point density and spacing are controled using spacing and growth factors set at curve end points. Surface grids are generated parametrically using the Advancing Front with Local Reconnection (AFLR) unstructured grid generation algorithm. A special 3D point insertion algorithm can improve grid quality on some surfaces. The volume grid generator runs as a seperate process, however this is transparent to the user. When the volume grid generator finishes its run, all the user need do is go to the grid visualization panel and press the load volume grid button. This will allow the user to view quality measures of the volume grid from within SolidMesh.

The code has an excellent user interface which consists of a single window with a graphics widget, message widget, group list widget, and an application dependent function panel. Function panels exist for CAD, topology, grid generation, visualization, file I/O, and solver setup (the solver setup panel is for MSFENS and not generic). The various panels are selected via buttons along a side bar. The interface is very clean and easy to follow. Function panels are arranged in a fashion that suggest the flow of the grid generation and solution setup process. The grid generation and visualization panels are missing the myriad of options that make many grid generation GUIs confusing; however this does not significantly effect the overall utility of the code.

There are many time-saving features available. The most obvious and probably most significant of these is the modaless interface. To perform an operation, the user selects the input for the desired operation and then clicks on the button to envoke it. This eliminates the need to prompt the user for input and drastically reduces the amount of time required to complete a particular operation. In the hands of an experienced user, operations can be performed very efficiently and with relatively few mouse-clicks and keystrokes. Another important user interface feature is overloaded function buttons. Most operations within the code behave differently depending on the type of input given. In most cases this allows the user to do the desired operation in the most efficient manner under the given circumstances. For example, a line can be created between two existing points, or from one point, a direction vector, and a distance, or a series of line segments can be created joining two or more selected points; all using differing inputs to a single function button.

Reviewers Comments:

- User interface is clean and easily navigated.
- Many time saving features and short cuts for the experienced user.
- User interaction paradigm is consistent throughout.
- Grid algorithms are very fast and efficient.
- Good grid visualization/quality measure features.
- Internal CAD system allows imported geometry to be repaired and/or modified. A complete geometry database can be created from scratch.
- The availability of the code and a pricing structure have not been set.
- No fomal user support structure exists, although current users may contact the developers directly.
- 2D capability through command line flag that enables 2D mode.
- Can achieve excellent grid quality.

A.7 Evaluation Summary for GridPro

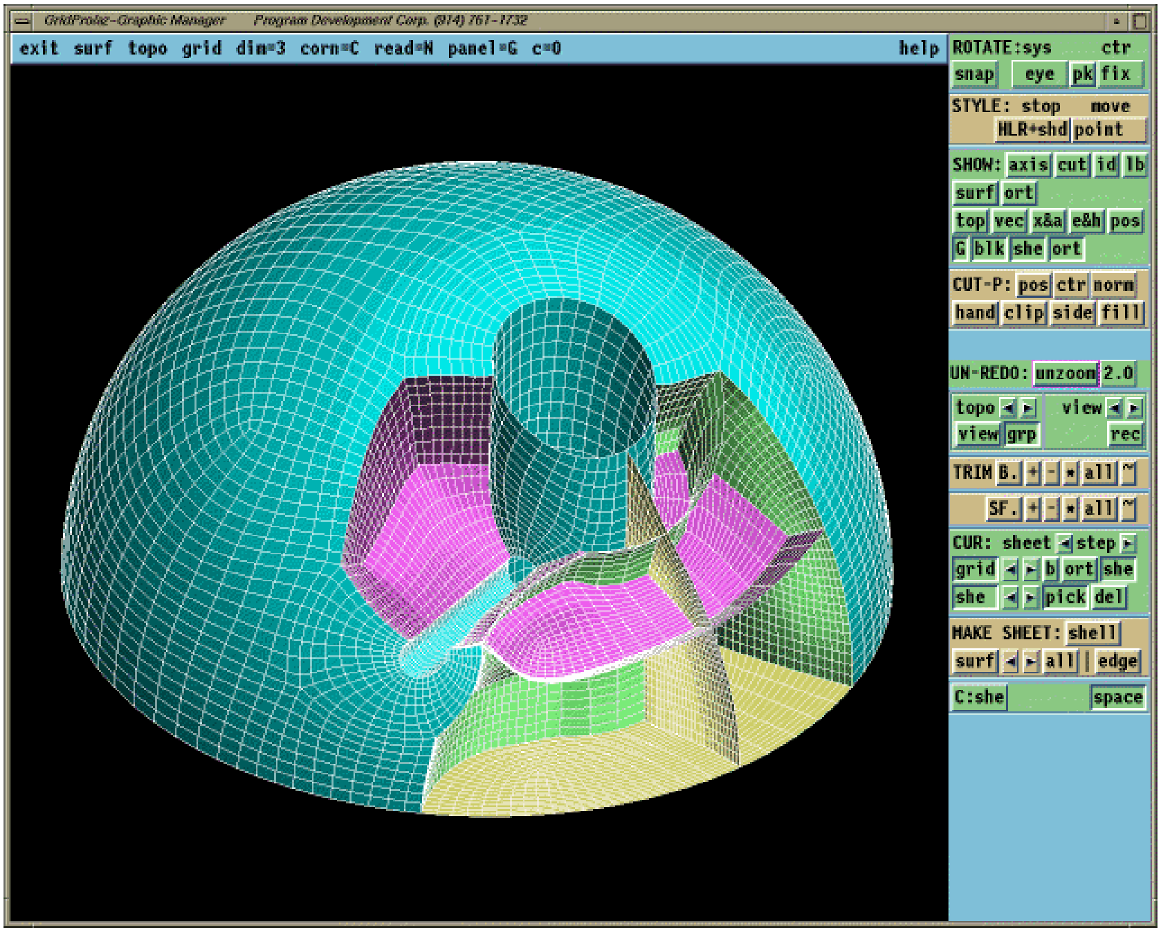

Figure 5: GridPro Interface and Sample Grid

Software Title: GridPro
Version: 3.0
Vendor: Program Development Corporation
Supported Platforms: SGI, IBM, DEC, HP, PC(Win95/NT)
Availability: Commercial Product
Contact:

Peter Eiseman

Phone: (914) 761-1732
Toll-free:
Fax: (914) 761-1735
Email: gridpro@gridpro.com
URL:

```
Documentation Provided: User Manual after attending training
User Support Hotline: Yes

Reviewer(s): Brian Jean

Grid Types: Block Structured Hexahedral
Grid Input Formats: Plot3D
Grid Output Formats: Plot3D, Native GridPro, NSC/NASTRAN
Geometry Input Formats: Proprietary.  NOTE: There are several utility
                        programs provided that translate common formats, such
                        as IGES, into PDC format.  IGES, NSC/NASTRAN, Quadrilateral Grid,
                        Triangular Grid, Mixed Element Grids are supported .
Geometry Output Formats: Proprietary

Supported Analysis Codes: Star-CD, CFX, Fluent, Fidap, TASCflow,
                          FINE/TURBO, TLNS3D, CFL3D

------------------------------------------------------------------------------------------
```

See Appendix B for an explaination of the items below

```
Automation Level:                 Moderate to High
Seat Time for Infrequent User:    Moderate
Seat Time for Frequent User:      Moderate to Low
User Interaction Paradigm:        Average
UserInterface Type:               GUI/Modal + batch code
Learning Curve:                   Hard
Documentation Quality:            Average
User Support Quality:             Good

Overall Capability:               Good
Grid Quality:                     Very Good
Curve/Surface Grid Fidelity:      Good
Large Data Set Capability:        Good
Stability/Maturity:               Average

Support for MPP Environment:      None
Grid Editing Features:            Extensive grid editing features for
                                  clustering, parametric design changes,
                                  and insertion/deletion of geometry.

Quality Measure Evaluation:       Skew, Aspect ratio, and Jacobian check

------------------------------------------------------------------------------------------
```

Code Summary:

The GridPro software consists of the main grid generator (a batch code), a GUI, and about twenty utility programs. The main code is controled by the GUI or may be run seperately in batch mode. In either mode, the main code is controled by a script written in Topology Input Language (TIL) code. The GUI is very useful, but quite hard to learn. It is currently supported on several popular architectures (see above).

The code has very powerful domain decomposition (blocking) and grid generation functions. There is no support for partial face matching in the blocking structure -only elementary (full face matched) blocks are supported. The abscense of partial face matching support may cause problems for some analysis codes, but because of the advanced blocking features it does not cause a problem building the blocking structure. Many time saving block construction features greatly reduce the time required to block out a volume grid. For example, the blocking structure for the grid pictured above was built in about ten minutes. The entire grid took about 20 minutes to generate (including time to build the geometry). Upon output, the code has an option to merge blocks and thereby reduce the total number of blocks the analysis code must deal with.

Excellent point spacing and grid smoothing functions are available. Unlike most other structured grid codes, point spacings are specified on surfaces and not on edges. A point spacing specification consists of an intial spacing off the surface and a stretching ratio for expansion into the field. The grid smoother produces the smoothest grids of any hex generator tested. One very nice feature of GridPro is the ability to smooth a very coarse (Euler) grid for speed and then come back and add points to produce the fine (viscous) grid. When applying point spacings, GridPro can be set to automatically enforce multi-grid restrictions on the dimensions of the grid.

The scripting language, which drives the batch code, is amoung the most powerful grid generation languages every defined. Features included a structured programming paradigm, parameterized subroutines which can be reused for different geoemtries, and global parameter definitions which allow the user to quickly change geometry, grid dimensions, grid spacings, etc. The availablility of the topology input language combined with the batch code might allow a user to build and smooth a coarse grid on a workstation and then add more points later, via the script, and produce the final grid on a HPC system.

The major weakness of GridPro is the lack of geometry preprocessing capability. In an environment where input is received from CAD systems, seperate geometry preprocessing software may be required if GridPro is to be used effectively.

Reviewers Comments:

- The GUI has room for improvement.
- Best smoothing algorithms I've seen.
- Excellent control over point spacing.
- Time to generate a grid is low compared with other GUI-based codes.
- The scripting language (TIL) is fairly easy to learn and very powerful.
- Geometry pre-processing capability is weak.
- Overall a very powerful system, once the geometry has been prepared.

A.8 Evaluation Summary for TrueGrid

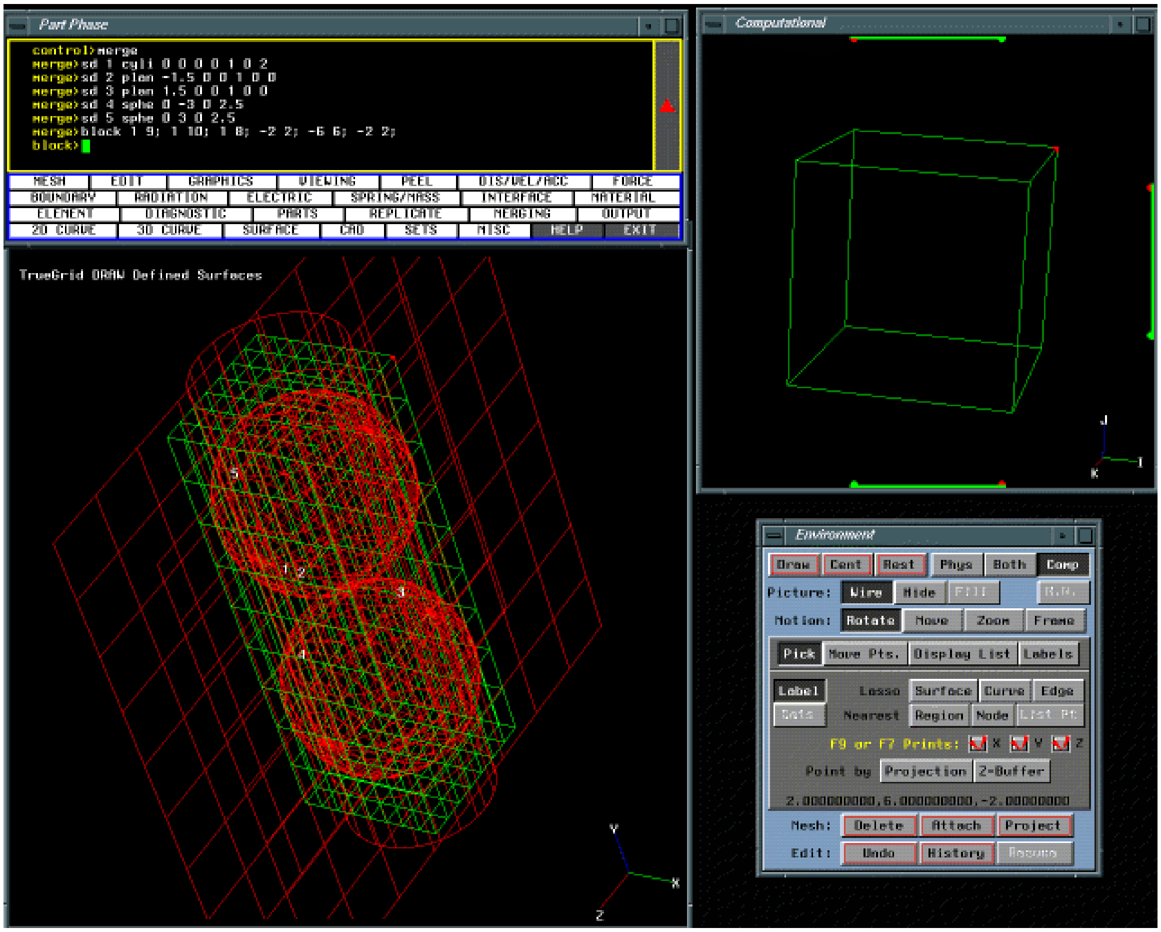

Figure 6: TrueGrid Interface and Sample Grid

```
Software Title:   TrueGrid
Version: 1.3.24
Vendor: XYZ Scientific Applications Inc.
Supported Platforms: DEC, SGI, IBM (AIX), HP, SUN, Windows NT/95, MSDOS 4.0
Availability: Commercial Software
Contact:
            XYZ Scientific Applications, Inc.
            1324 Concannon Blvd.
            Livermore, CA 94550

            Phone: (510) 373-0628
            FAX: (510) 373-6326
            email: info@truegrid.com
            URL: http://truegrid.com
```

```
Documentation Provided: Users Manual and Tutorial

User Support Hotline: Yes

Reviewer(s): Richard Weed

Grid Types: Structured and Unstructured Hexahedral Meshes for CSM and CFD

Grid Input Formats: IGES
Grid Output Formats:

            CSM Codes:  ABAQUS, ALE3D, ANSYS, AUTODYN, DYNA3D, LS-DYNA3D,
                        LS-NIKE, MARC, NASTRAN, PATRAN Neutral File, TOPAZ3D
            CFD Codes:  CFD-ACE, CFX, COMPACT, FIDAP, FLUENT, GRIDGEN3D,
                        NEKTON, PLOT3D, REFLEQS, STAR-CD, TASCflow


Geometry Input Formats: IGES
Geometry Output Formats: See Grid Output

Supported Grid/Analysis Codes: See Grid Output Formats

------------------------------------------------------------------------------------------
```

See Appendix B for an explanation of the items below

```
Automation Level:               High
Seat Time for Infrequent User: Medium
Seat Time for Frequent User:   Low to Medium
User Interaction Paradigm:     Average
User Interface Type:           GUI Directed Scripting, GUI based surface
                               selection and manipulation

Learning Curve:                Low
Documentation Quality:         Good
User Support Quality:          Good
Overall Capability:            Good
Grid Quality:                  Good
Curve/Surface Grid Fidelity:   Good
Large Data Set Capability:     Good
Stability/Maturity:            High

Support for MPP Environment:   Unknown
Grid Editing Features:         Interactive GUI based editing as well
                               as ASCII script files
Quality Measure Evaluation:    Orthoganality checks

------------------------------------------------------------------------------------------
```

Code Summary:

TrueGrid is a GUI based hexahedral grid code for CSM and CFD applications. TrueGrid is a descendent of the INGRID hexahedral grid code developed at Lawrence Livemore National Labs to support its suite of CSM codes (DYNA3D, NIKE3D, TOPAZ). Unlike INGRID, TrueGrid has a modern GUI based interface that is more approachable for the novice or infrequent user. However, its underlying input paradigm is text based script commands whose syntax and function follow the command structure and function of INGRID. This allows experienced INGRID users to transition to TrueGrid with a moderate amount of training. All input commands can be entered directly into the command window or via dialogue boxes that prompt for the proper input. This mixture of GUI and scripting combines the best features of both and would be welcome addition to other more heavily GUI based codes.

TrueGrid's major strengths are its native support for specifying boundary conditions and material properties for a variety of CSM and CFD codes as well as its powerful projection methods for placing mesh faces, edges, surfaces and nodes onto the desired surface and curve geometries. This technology allows surfaces and mesh boundaries to be defined independently. The projection methods will automatically define the appropriate interfaces and common edges of surfaces. In addition, script files can be saved for each session that are easily edited to include new features or delete old ones.

A typical TrueGrid session requires the user to pass through different predefined phases to generate the final grid. The syntax for these phases is similar to that used in INGRID. In each phase, input options can be chosen from the main menu. The Part Phase is used to define the mesh by appropriate positioning, smoothing, zoning, etc. of predefined parts of the mesh. In the Merge Phase, parts are merged into one model. Boundary conditions and material properties are defined in the Parts Phase. TrueGrid provides functions for node redistribution and spacing as well as elliptic smoothing. Mesh quality can also be evaluated for orthoganality etc.

TrueGrid supports import of predefined IGES entities to define initial surface geometries. Although primarily focused at CSM applications, TrueGrid also supports export to a variety of CFD codes and commercial Finite Element solvers.

Reviewers Comments:

- Only a partial evaluation was completed for this code.
- TrueGrid is an excellent grid generation tool for anyone requiring quality meshes for CSM codes such as DYNA3D.
- The mixture of scripting and GUI based inputs in TrueGrid provides a convenient and easy to learn interface. Though not as powerful as other more heavily GUI based codes, the experienced user can, in most cases, generate an equivalent quality grid in fewer steps.
- Our experience with TrueGrid is limited to support for CSM applications, its utility as a grid generator for CFD applications still remains to be determined.

A.9 Evaluation Summary for INGRID

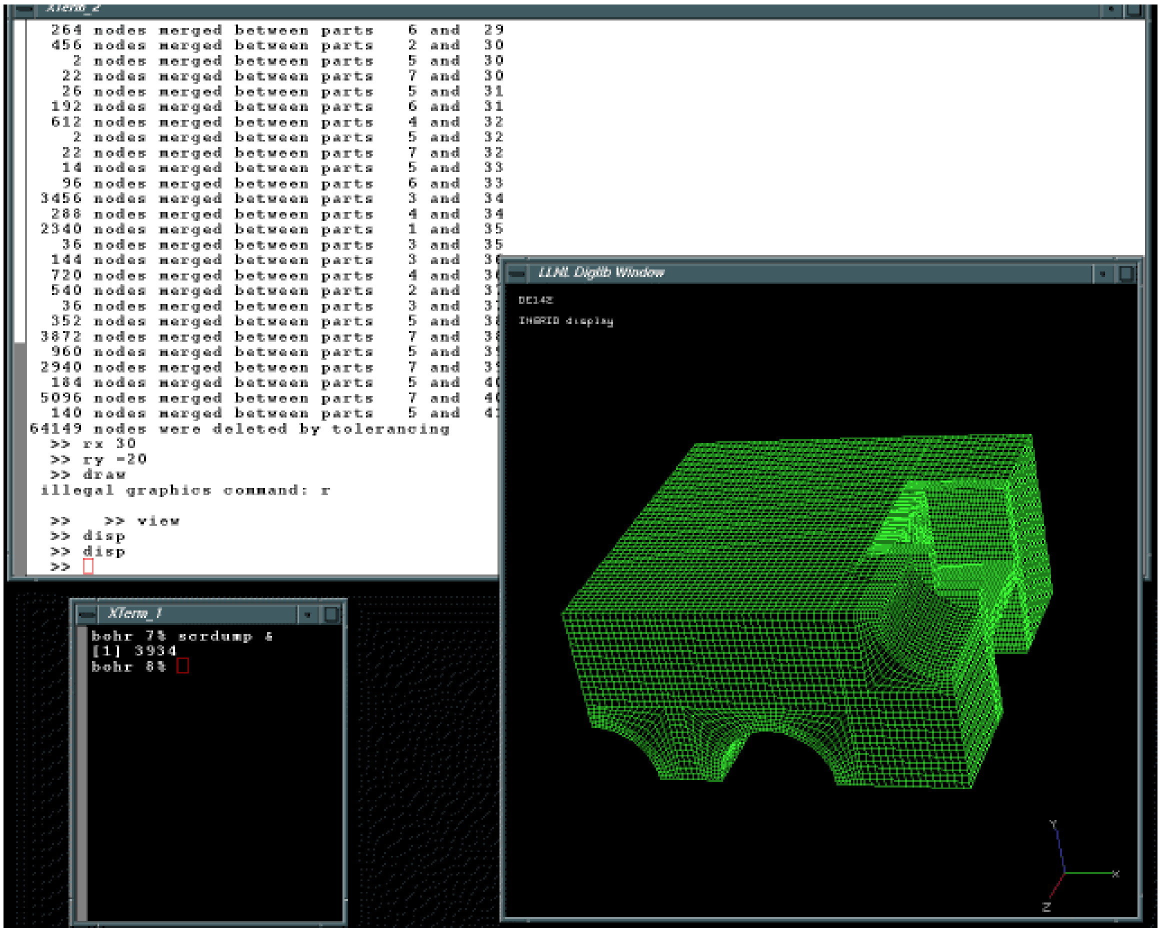

Figure 7: INGRID Interface and Sample Grid

```
Software Title:   INGRID
Version: 1996e
Vendor:  Lawrence Livermore National Labs
Supported Platforms: Cray C90, SGI Origin 2000
Availability: Public Domain - Distributed to approved U.S. industries and universities
Contact:

          Industrial Partnerships and Commercialization
          Lawrence Livermore National Laboratory, Mail Stop L-795
          7000 East Avenue
          Livermore, CA 94551
```

Contacts

Peter J. Raboin	Jerry Lin
Phone: (510) 422-867	Phone: (510) 423-0907
Fax: (510) 422-2085	Fax: (510) 422-2085
E-mail: raboin1@llnl.gov	E-mail: lin5@llnl.gov
Mail code: L-122	Mail code: L-122

Documentation Provided: Users Manual

User Support Hotline: No

Reviewer(s): Richard Weed

Grid Types: Unstructured Hexahedral Grids for CSM Codes (DYNA3D, NIKE3D, TOPAZ3D)

Grid Input Formats: Script commands
Grid Output Formats: DYNA3D, LS-DYNA3D, NIKE3D, LS-NIKE, TOPAZ3D, ADINA, ANSYS

Geometry Input Formats: Geometry definition is by script commands or edge files
Geometry Output Formats: See Grid Output

Supported Grid/Analysis Codes: See Grid Output Formats

See Appendix B for an explanation of the items below

Automation Level:	Medium
Seat Time for Infrequent User:	High
Seat Time for Frequent User:	Medium
User Interaction Paradigm:	Average
User Interface Type:	Script based ASCII input files, some interactive editing capability. Digilib graphics library for viewing output.
Learning Curve:	High
Documentation Quality:	Good
User Support Quality:	Good
Overall Capability:	Good
Grid Quality:	Average
Curve/Surface Grid Fidelity:	Good
Large Data Set Capability:	Average (Some applications requires parts to be generated separately and merged manually)
Stability/Maturity:	High
Support for MPP Environment:	Unknown
Grid Editing Features:	ASCII scripts can be edited. Some interactive editing from command line

```
Quality Measure Evaluation:    None
```

Code Summary:

INGRID is a script based hexahedral grid generation code for CSM applications. INGRID was developed at Lawrence Livermore National Labs to support its suite of structural dynamics codes. The primary focus of INGRID are the DYNA3D and NIKE3D codes. INGRID's primary strength is its native ability to generate the required boundary conditions and material definitions used in the DYNA3D and NIKE3D codes. Being script based, it has no mouse driven graphics interface. INGRID has a 3D graphics capability for viewing meshes. Common graphics task such as zooming, changing viewing angle, etc. are controlled by script commands. INGRID possess some of the projection technology found in TrueGrid.

In the hands of an experienced user, INGRID will generate good quality grids for its target analysis codes. However, there is a large learning curve associated with becoming proficient with the code. The script base approach makes modifying an existing grid or creating a new one from a previous script file a matter of editing the scripts. A more interactive approach to defining the various boundary conditions between parts would greatly reduce the time required to set up a DYNA3D run.

Reviewers Comments:

- Only a partial evaluation was completed for this code.
- INGRID will produce acceptable grids for most problems. The major deficiency of the code is there is no interactive way of specifying boundary conditions between parts (ie., tied node sets, slide lines, etc.)
- The script based approach has a high learning curve for the casual user. The ability to edit the script file for a previous case or configuration helps in modifying and editing grids.

A.10 Evaluation Summary for GRIDTOOL

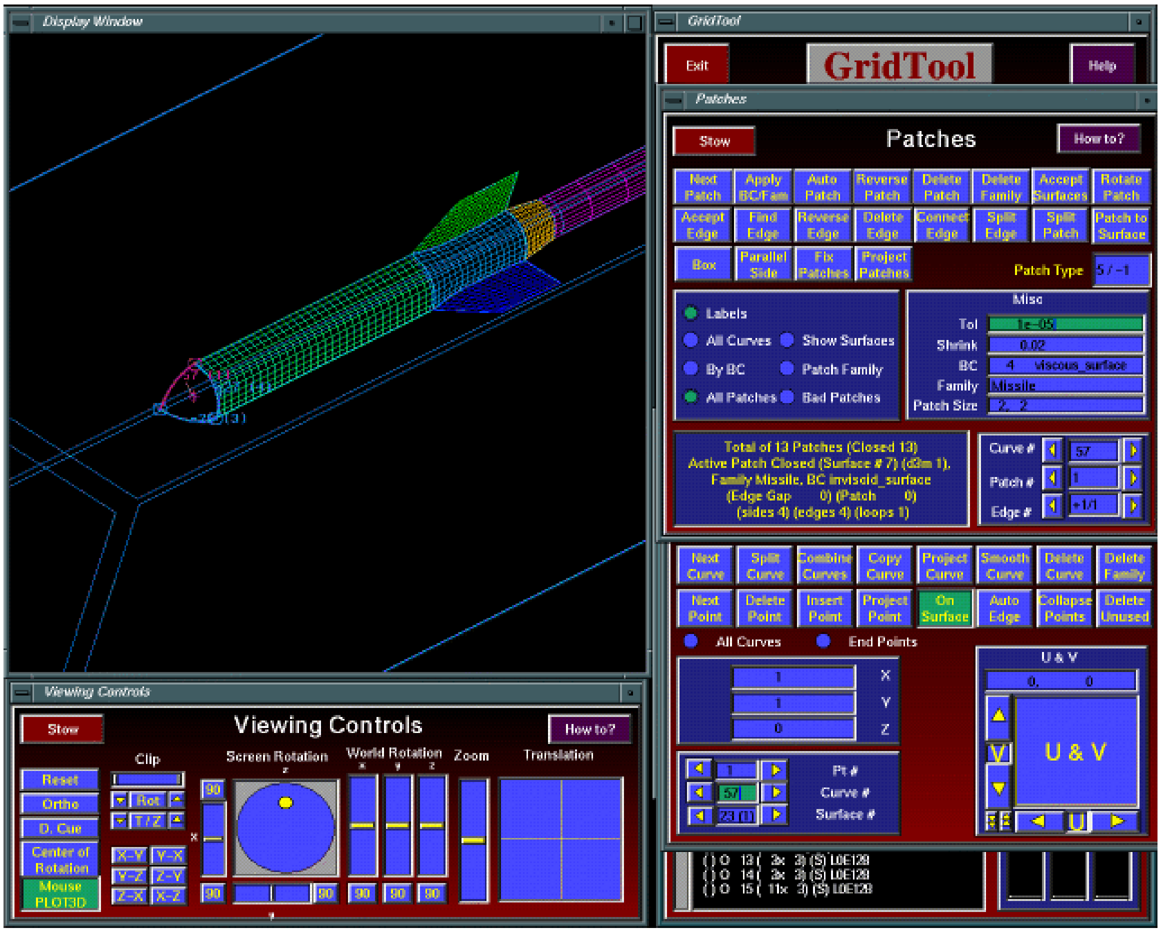

Figure 8: GRIDTOOL Interface and Sample Grid

```
Software Title: GridTool
Version: 3.2
Vendor: NASA Langley Research Center Geometry Laboratory
Supported Platforms: SGI
Availability: Public Domain - Distributed on approved request by NASA
Contact:

            Ms. Pat Kerr
            NASA Langley Research Center
            MS 125
            Hampton, VA  23681-0001

            Phone: (757)864-5782
            FAX:   (757)864-8910
```

```
             URL: http://geolab.larc.nasa.gov/GridTool/

Documentation Provided: No formal users manual - There is a Web based
                        tutorial and code description.
User Support Hotline:

Reviewer(s): Richard Weed

Grid Types: Surface Geometry and Boundary Definition

Grid Input Formats: Native restart, IGES, PLOT3D, GRIDGEN, VGRID, FELISA,
                                   LaWGS
Grid Output Formats: Native restart, IGES, PLOT3D, GRIDGEN, VGRID,
                     FELISA, LaWGS

Geometry Input Formats: See Grid Input
Geometry Output Formats: See Grid Output

Supported Grid/Analysis Codes: VGRID, FELISA, USM3D

------------------------------------------------------------------------------------------
```

See Appendix B for an explanation of the items below

```
Automation Level:               High
Seat Time for Infrequent User: High
Seat Time for Frequent User:   Medium to High (depends on complexity
                               of original CAD or other defining
                               geometry)
User Interaction Paradigm:     Average
User Interface Type:           GUI

Learning Curve:                Moderate
Documentation Quality:         Poor
User Support Quality:          Good

Overall Capability: Good (with proper training)
Grid Quality: Good
Curve/Surface Grid Fidelity: Good
Large Data Set Capability: Good
Stability/Maturity: Medium

Support for MPP Environment:   Unknown
Grid Editing Features:         Surface Point Redistribution Functions,
                               Patch/Curve Editing
Quality Measure Evaluation: Automatic checks for valid patches

------------------------------------------------------------------------------------------
```

Code Summary:

GridTool is an interactive tool designed to bridge the gap between the definition of surface geometry via a CAD system and structured and unstructured grid generation packages. Current support is aimed at the two unstructured codes (VGRID and FELISA) used in NASA Langley's Tetrahedral Unstructured Software System (TetrUSS). GridTool provides a GUI based front end for VGRID that allows the interactive definition and manipulation of surface patches, NURBS based surfaces, point distributions, boundaries, and boundary conditions from an initial geometry definition obtained from a CAD system or other structured grid generation codes. GridTool is also used to define the background , source locations, and initial front spacing used by VGRID in the Advancing Front and Advancing Layer unstructured grid generation schemes. Finally, the initial front generated by VGRID is used to define a triangulated surface definition used by VGRID in the final volume grid generation. The surfaces are defined by projection of the initial front generated by VGRID on to the CAD surfaces.

The GridTool GUI consists of a viewing port, a main control panel, and mouse selected subpanels that access functions for I/O, points and curves definition, patch definition and editing, background grid and grid source definition, etc. An initial geometry definition is required. GridTool supports import of IGES entities such as NURBS surfaces, trimmed surfaces, parametric splines and surfaces, and curves on a parametric surface. In addition to IGES formats, GridTool will also import surfaces defined in GRIDGEN, PLOT3D, LaWGS, or VGRID-NET formats.

A typical GridTool/VGRID session will start with the user inputting the initial defining geometry. The curves or surfaces from this geometry are then used to define families of parametric curves or patches. Selection of of geometric entities such as curves or surfaces can be accomplished by key-board input for each patch or interactively by positioning the mouse in the viewing screen and pressing an appropriate "hot key". In conjunction with the definition of a surface patch, the user defines the appropriate flow solver boundary conditions for that particular patch. Once a patch has been defined, it can be converted to a NURBS surface and projected to coincide with the original CAD geometry.

After definition of the surface patches and boundary conditions, a background grid is generated for use by VGRID in the AFM or ALM schemes. This step requires the user to define grid "source" points and lines that control the initial spacing for the advancing front or layer. After definition of the background grid, an initial VGRID .d3m file is saved. VGRID is then used to generate the initial front. This initial front is read back into GridTool and projected on the defining NURBS surfaces. GridTool can then be used to redistribute points. The output from this second pass through GridTool is used by VGRID to generate the final volumetric grid system.

In the hands of an experienced user, GridTool can greatly reduce the time to go from an initial CAD geometry to an unstructured grid. However, the lack of detailed documentation makes dedicated training a must. Once mastered, the interface is relatively easy to navigate and the function of most of the controls in the subpanels becomes intiutive. The on-line documentation will provide enough information for someone familiar with basic CAD and grid generation concepts to figure out with a little experimentation how to generate the required surface patches. The generation of the background grids and placement of the grid "source" terms is poorly explained and is the most confusing step in the process.

Reviewers Comments:

- GridTool would be an excellent tool for surface geometry definition if more adequate documentation was provided.
- GridTool was extremely slow on the systems we installed it on. It got even worse when trying to run in a remote X session.
- On the sample problem we tried, GridTool kept complaining about bad patches even though it reported all patches were closed and oriented in the same direction. It wasn't determined if this is a bug or a feature.
- The generation of the initial front needs to be imbedded in GridTool.
- Going back and forth between VGRID and GRIDTOOL is annoying.

A.11 Evaluation Summary for VGRID

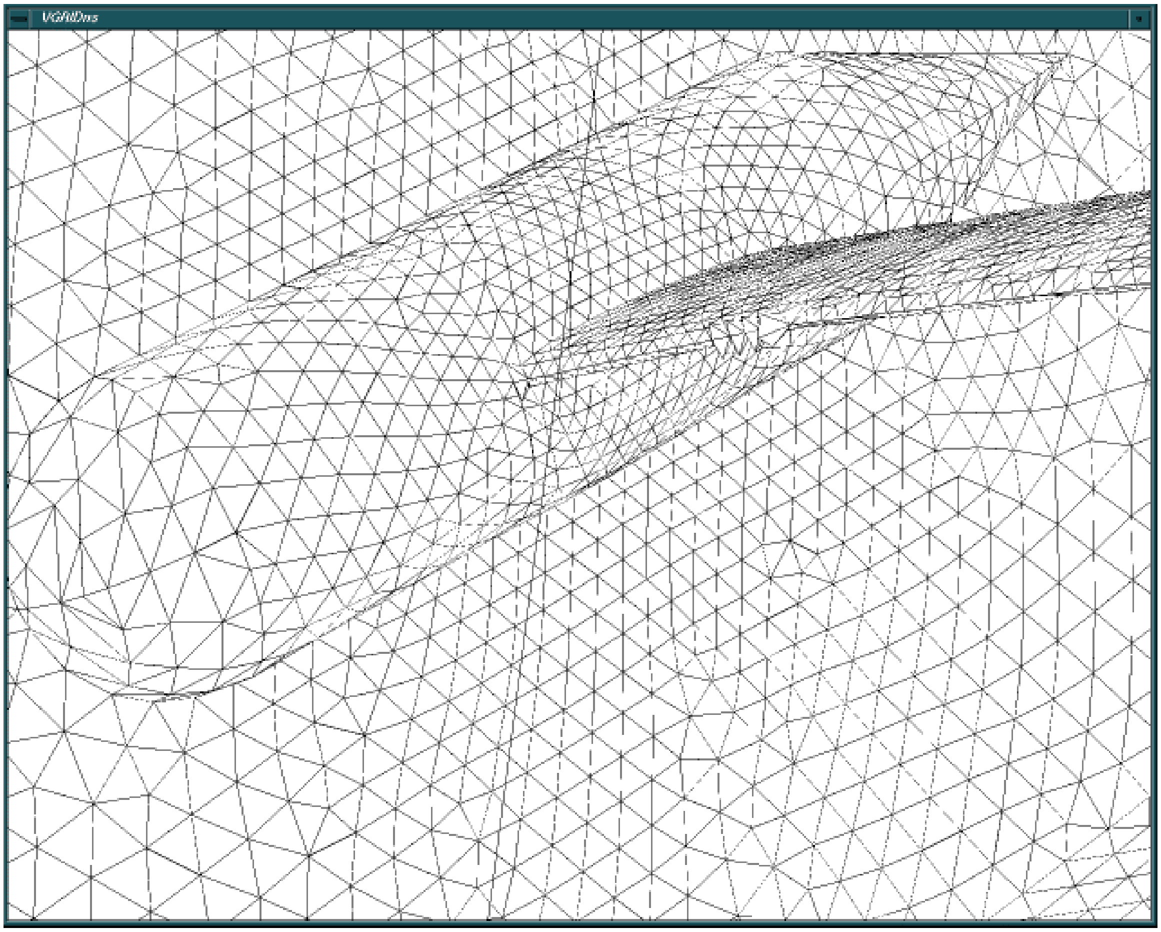

Figure 9: VGRID Sample Grid

```
Software Title:   VGRID, VGRIDns
Version:
Vendor: ViGYAN, Inc
Supported Platforms: SGI
Availability: Public Domain - Distributed on approved request by ViGYAN
Contact:

            Dr. Shahyar Pirzadeh
            NASA Langley Research Center
            MS 499
            Hampton, VA 23681-0001

            Phone: (757)864-2245
            FAX: (757)864-8469
```

```
            E-mail: s.pirzadeh@larc.nasa.gov
            URL: http://www.vigyan.com/vigyan/vgrid.html

Documentation Provided: No formal users manual - There is a Web based
                        tutorial and code description.
User Support Hotline:

Reviewer(s): Richard Weed

Grid Types: Unstructured Tetrahedral CFD meshes

Grid Input Formats: Native .d3m formats
Grid Output Formats: Native .grd format used by USM3D

Geometry Input Formats: See Grid Input
Geometry Output Formats: See Grid Output

Supported Analysis Codes: USM3D

-----------------------------------------------------------------------------------------
```

See Appendix B for an explanation of the items below

```
Automation Level:               High
Seat Time for Infrequent User:  Medium
Seat Time for Frequent User:    Low to Medium
User Interaction Paradigm:      Average
User Interface Type:            3D display of geometry

Learning Curve:                 Moderate
Documentation Quality:          Poor
User Support Quality:           Good

Overall Capability:             Good (with proper training)
Grid Quality:                   Good
Curve/Surface Grid Fidelity:    Good
Large Data Set Capability:      Good
Stability/Maturity:             Good

Support for MPP Environment:    Unknown
Grid Editing Features:          Interactive rearrangement of surface patches.
                                Local remeshing done by POSTGRID post-processing
                                tool.
Quality Measure Evaluation:     Surface Grid Quality. Volume grid quality checks
                                performed in POSTGRID.
-----------------------------------------------------------------------------------------
```

Code Summary:

VGRID is a standalone tool for generating unstructured tetrahedral meshes for CFD analyses. VGRID is an integral component of the NASA Langley TetrUSS software system. VGRID provides an interactive environment for generating and displaying surface and volume unstructured tetrahedral grids. VGRID provides the capability for the fast and convenient generation of grids about complex geometries. Current versions of VGRID include the base version (VGRID) that is used primarily for Euler analysis and a new version in beta testing designated VGRIDns that is used for generating viscous grids. The underlying grid generation scheme in VGRID is the Advancing Front Method (AFM) which marches grid points out from a predefined surface in a manner similar to the hyperbolic and parabolic schemes used in structured mesh generation. VGRIDns uses the similar Advancing Layer Method (ALM) to generate tetrahedral grids from a reference prismatic grid.

VGRID is normally used in conjunction with the GridTool GUI based geometry definition tool. GridTool and VGRID are used to generate an initial front to provide a triangulated surface grid that is used in the volumetric grid generation. A .d3m file is read into VGRID (VGRIDns). Repeated pressing of the ESC key brings up mouse driven menus that control different viewing options. The actual grid generation is done automatically. A separate post-processing tool, POSTGRID, is provided to perform local remeshing and to access grid quality.

The lack of documentation on the operation of VGRID makes formal training a must. However, the code is in general use at NASA and throughout the aerospace community and is considered a standard tool for unstructured grid generation. The code appears to be very stable and robust for inviscid grid generation.

Reviewers Comments:

- This evaluation should be considered incomplete at this time until we get a chance to get some formal training.
- Attempts to run the sample cases "blind" revealed the code to be fast and responsive.

APPENDIX B.
DEFINITIONS AND TERMS USED FOR CODE EVALUATION

B.1 Automation Level

Automation level is an aggregate measure of several user interaction required, time saving features, and the level of "artificial intelligence" incorporated into the code.

```
Very Low:  Operations using the code are, in general, completely
           manual, offering no automation and few, if any, time-saving
           features.

Low: Operations are, in general, completely manual with no automation
     but some time saving features.

Medium: Most operations are manual.  Some operations are automatic and
        a good number of time-saving features are available.

High: Sevaral operations are manual.  Many operations are automatic and
      a large number of time-saving features are available.

Very High: The code is largely automatic.  Very few if any operations
           are manual.
```

B.2 Seat Time

Seat time is a measure of wall-clock time required for a user to generate a grid. Seat time is directly related to automation level above, but it also includes factors such as ease of use and learning curve. Two categories are given for this measure, one for the frequent user and one for the infrequent user. The frequent user is one who is very familiar with the respective code and seldom if ever refers to the code documentation. The infrequent user is one who is familiar with only the basic code operation and must refer to the manual regularly.

B.3 User Interaction Paradigm

User interaction paradigm is a subjective measure user interface intuitiveness, consistency, and ease of use.

```
Poor: The code is difficult to use.  Commands and operations have no
      consistency in their format or syntax.  The code may seem
      confusing and unpredictable to the novice user.  If applicable,
      navigation between code modules is cumbersome and no well defined
      process for generating a grid exists.

Average: The code is reasonably easy to use and is somewhat consistent
         in its behavior and command syntax.  Navigation between
         code modules and/or function panels is reasonably intuitive.
         The grid generation process is reasonably easy to follow.

Good: Behavior and/or command syntax is easy and intuitive.  Navigation
```

```
between modules and/or function panels is intuitive and easy
to master.  The grid generation process is well defined and
easy to follow.
```

B.4 User Interface Type

There are two basic user interface types, graphical user interface (GUI) and script or language based interfaces.

B.5 Learning Curve

Learning curve is a relative measure of the time and/or effort required to become an intermediate to advanced user of the code.

```
Easy:  The code is fairly simple to learn using only the documentation
       provided and user support hotline, if available.  Concepts and
       procedures are well defined and consistent.  A training course
       may be beneficial, but is not necessary.

Moderate: The code is difficult to learn with the provided documentation
          and support.  Concepts and procedures may be difficult to
          master.  A training course is recommended.

Hard: The code is very difficult to learn without a good training
      course.  Concepts and proceedures are esoteric and very
      difficult to master.  A training course is necessary in order
      to effectively use the code.
```

B.6 Documentation Quality

This item reflects the quality and quantity of documentation provided by the vendor. This includes installation manuals, user manuals, references manuals, and tutorials.

```
Poor:  Little or no documentation is provided.  Material that is
       provided is of poor quality and is not well organized.  Help
       on specific topics is non-existent, virtually impossible to
       find, or of poor quality.

Average: Adequate documentation for the average user is provided.
         Overall organization of the material is good.  Help on
         specific topics may difficult to locate or not completely
         explained.

Good: Documentation is extensive, well organized, and polished.
      Help on specific topics is easy to find and the material is
      thoroughly explained.
```

B.7 User Support Quality

This item reflects the quality and/or accessability of user support via email or telephone.

Poor: User support is of poor quality. Support personel are difficult
to contact, lack thorough knowledge of the code, and/or are
unresponsive.

Average: Support personel are easily contacted and have a reasonable
knowledge of the code. Questions are answered within a
reasonable amount of time and answers are adequate in most
cases.

Good: Support personel are easily contacted and have a thorough knowledge
of the code. Questions are answered immediately in most cases.
Support personel are willing to walk the user through examples
and work with the user's data when needed.

B.8 Grid Quality

Grid quality reflects several characteristics of the code such as the ability to control grid spacing, orthogonality, smoothness, stretching and accuracy.

Poor: The user has very little control over the grid characteristics
listed above.

Average: The user has adequate control of grid characteristics and is
able to routinely generate acceptable grids.

Good: The user has very fine control of grid characteristics and is
able to routinely generate high-quality grids.

B.9 Curve/Surface Grid Fidelity

Fidelity is a measure of the ability of the code to generate a grid that conforms to given curves and/or surfaces supplied from an external source such as Pro-Engineer or Unigraphics. Some codes use given geometry directly, while others project grids to curves and surfaces, and still others rebuild the geometry to conform with their own internal geometry database.

Poor: Grid points are generally not exactly on the given surfaces.
Grid points may be allowed to float off of surfaces, may
be projected inaccurately, or the geometry may have to be
rebuilt by the user or the code introducing errors. Projection
routines, if provided, are not robust. Geometry representation
within the code may by discrete and not account for the presence
of analytical surfaces such as NURBS.

Average: Grid points, in general, lie on the given surfaces. Projection
routines, if present, are generally robust, but may have problems
on sculpted surfaces or in areas of hight curvature. Grid
smoothing routines may move grid points slightly off of surfaces.
Geometry representation within the code allows for both discrete

```
          and analytical surfaces.

    Good: Grid points are almost always on the given surfaces.  Projection
          routines, if present, are robust and accurate.  Some grids may
          be generated and smoothed in the parameter space of given
          surfaces.  Grid smoothing always keeps points exactly on the
          given surfaces.  Geometry representation allows for both discrete
          and analytical surfaces.
```

B.10 Support for MPP Environment

This is a list of features in the code that are useful in a parallel computing environment.

B.11 Grid Editing Features

Grid editing allows the user to change an existing grid system. Features may include snapping a grid to a slightly different geometry, allowing parametric changes to given geometry/grids, or automatically replacing existing geometry with new geometry.

B.12 Quality Measure Evaluation

This item indicates the tools available within the code for measuring the quality of a grid. Features may include histograms or shaded rendering of quatities such as skew angle, aspect ratio, stretching, and/or off-boundary spacing.

APPENDIX C.
WORKSHOP ATTENDEES

The following list of people attended the Grid Generation Capabilities Enhancement Workshop held at the University of Texas in Austin on February 11-12, 1998.

```
N. (Radha) Radhakrishnan - CEWES MSRC
Louis Turcotte - CEWES MSRC
Raju Namburu - CEWES MSRC (representing Kent Kimsey)
Photos Papados - CEWES MSRC
David Richards - CEWES MSRC
Alan Stagg - CEWES MSRC
Rama Valisetty - CEWES MSRC
Brian Jean - CEWES MSRC

Joe McCaffrey - NAVO MSRC

Douglas Blake - ASC MSRC (representing Joe Shang)
Terrell Hand - ASC MSRC (AEDC)

Sandy Landsberg - NRL (representing Jay Boris)
Ravi Ramamurti - NRL

Robert Cooper - LLNL (representing  Charlie Holland)

Patrick Knupp - Sandia
Tim Tautges - Sandia

Dick Pritchard - NRC (PET)
Wayne Mastin - NRC/MSU (PET)

Joe Thompson - MSU (PET)
Richard Weed - MSU (PET)
Steve Bova - MSU (PET)
Raghu Machiraju - MSU (PET)

Mary Wheeler - Texas (PET)
Clint Dawson - Texas (PET)
Victor Parr - Texas (PET)
Bob Fithen - Texas (PET)

Keith Bedford - Ohio State (PET)
Suxia Zhzng - Ohio State (PET)
Sean O'Neil - Ohio State (PET)
David Welsh - Ohio State (PET)

Yuping Zhu - Syracuse (PET)
```

LeRay Dandy - NCSA (PET)

Sriprakash Sarathy - Clark Atlanta (PET)

Bharat Soni - MSU (ASC/ARL PET)

Surya Dinavahi - MSU (ARL PET)

Jay Boisseau - SDSC (NAVO PET)

Graham Carey - Texas
John Kallinderis - Texas
Monica Martinez - Texas
Srinivas Chippada - Texas
Robert Kirby - Texas
Robert McLay - Texas